This implementation-oriented book provides a clear and concise presentation of how to apply fieldbuses for process control. Based on experience collected from end-users in a wide range of industries around the world, it provides "how-to" information for all phases of the system lifecycle from engineering to device and strategy configuration, installation, commissioning, troubleshooting, operation, and maintenance.

Fieldbuses for Process Control covers the three leading process fieldbus technologies: HART, FOUNDATION™ Fieldbus and PROFIBUS-PA. It covers both field-level and the Ethernet-based host-level networking. The text addresses concerns and solutions for interoperability, integration, and migration as well as availability and safety.

A chapter on benefits helps engineers justify business advantages to management. The final chapter provides an in depth explanation of how these fieldbus technologies work. The author exposes similarities, differences, and capabilities of each fieldbus technology.

Fieldbuses for Process Control is a must-have for system designers, control engineers and technicians. Process engineers will also find this very beneficial when learning about the capabilities of fieldbus technologies. It is ideal for both organized training courses and for self-study and will remain a handy reference when configuring and troubleshooting systems. This book is sure to be a well-thumbed addition to every control engineer's bookshelf.

ISA–The Instrumentation, Systems, and Automation Society
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, NC
27709 USA
Phone: (919) 549-8411
Fax: (919) 549-8288
Email: info@isa.org
ISA OnLine: www.isa.org

ISBN 1-55617-904-9
90000
9 781556 179044

Fieldbuses for Process Control:
Engineering, Operation, and Maintenance

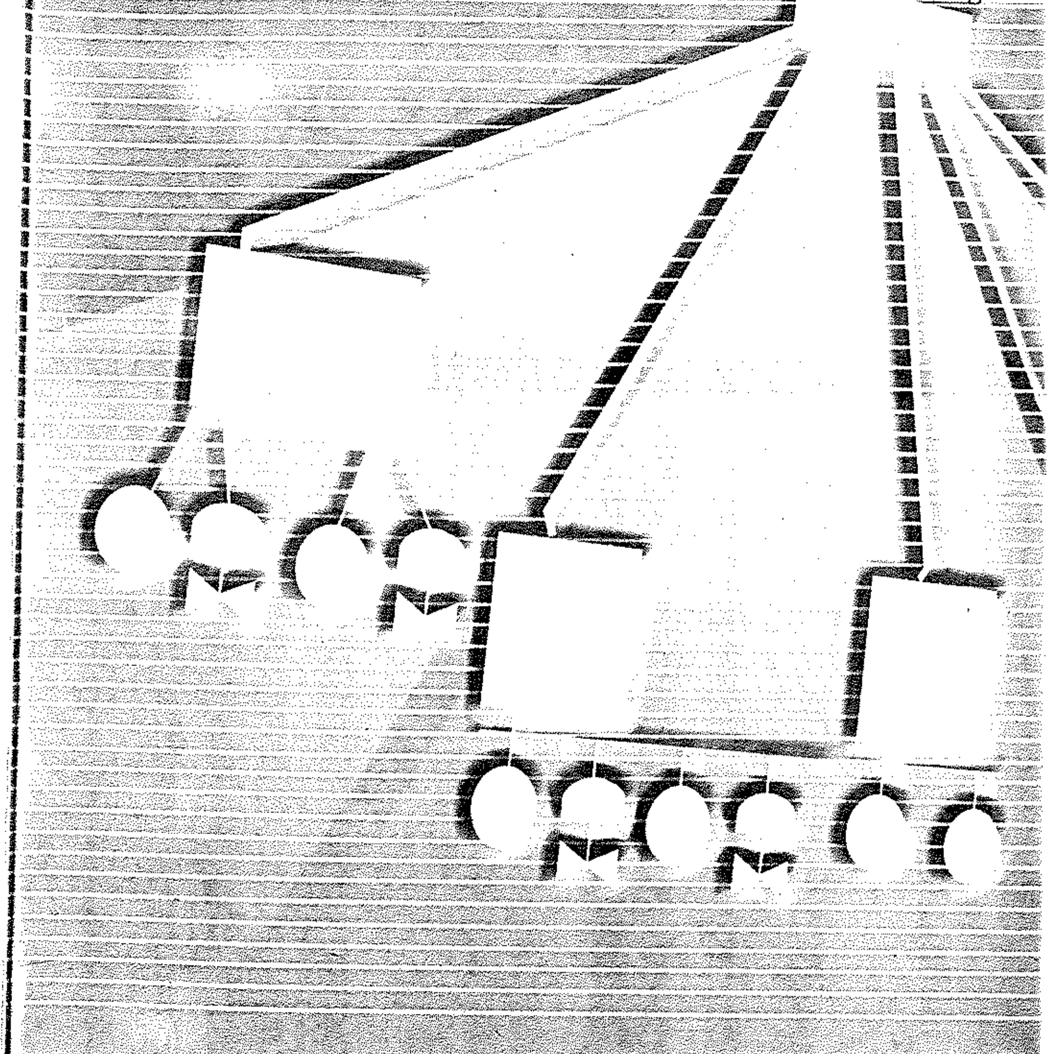
JONAS BERGE

ISA

Fieldbuses for Process Control:

Engineering, Operation, and Maintenance

Jonas Berge



ISA–The Instrumentation, Systems, and Automation Society

Fieldbuses for Process Control:

Engineering, Operation and Maintenance

Jonas Berge

ISA–The Instrumentation, Systems,
and Automation Society

Fieldbuses for Process Control:

Engineering, Operation and Maintenance

Contents

Notice

The information presented in this publication is for the general education of the reader. Because neither the author nor the publisher have any control over the use of the information by the reader, both the author and the publisher disclaim any and all liability of any kind arising out of such use. The reader is expected to exercise sound professional judgment in using any of the information presented in a particular application.

Additionally, neither the author nor the publisher have investigated or considered the affect of any patents on the ability of the reader to use any of the information in a particular application. The reader is responsible for reviewing any possible patents that may affect any particular use of the information presented.

Any references to commercial products in the work are cited as examples only. Neither the author nor the publisher endorse any referenced commercial product. Neither the author nor the publisher make any representation regarding the availability of any referenced commercial product at any time. The manufacturer's instructions on use of any commercial product must be followed at all times, even if in conflict with the information in this publication.

Any trademarks or tradenames reference belong to the respective owners of the mark or name. FOUNDATION™ Fieldbus is a registered trademark of the Fieldbus Foundation, HART® is a registered trademark of the HART Communication Foundation, SERIPLEX® is a registered trademark of Square D Company, Interbus-S is a trademark of the Interbus Club, DeviceNet is a trademark of the Open DeviceNet Vendor Association and ControlNet is a trademark of ControlNet International.

Copyright © 2002 ISA - The Instrumentation, Systems, and Automation Society

All rights reserved.

Printed in the United States of America.
10 9 8 7 6 5 4 3 2

ISBN 1-55617-904-9
First paperback edition.

No part of this work may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the publisher.

ISA
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, NC 27709

Library of Congress Cataloging-in-Publication Data

Berge, Jonas.
Fieldbuses for process control: engineering, operation, and maintenance / Jonas Berge.
p. cm.
Includes index.
ISBN 1-55617-904-9 (pbk.)
1. Intelligent control systems. 2. Process control. 3. Industrial electronics. I. Title.
TJ217.5.B47 2004
670.42'75--dc22

2004016957

Dedication

Till min pappa och till minne av min mamma

1

Introduction

The first process control systems were understandably analog, simple devices with signal formats that were essentially determined by the need for an architecture with a minimum number of costly CPUs. Networking was introduced into industrial automation in the 1970s and first utilized in direct digital control (DDC) systems between computer and I/O (input/output). Later, it was used in distributed control systems (DCS) and programmable logic controller (PLC) systems to connect the controllers and operator consoles. However, digital communications in smaller devices such as transmitters on the plant floor was not seen until the 1980s, and true communication bus networking of field instruments did not gain wide acceptance until the 1990s.

At the other extreme, corporations network their plants across the globe to the corporate headquarters via the Internet. The coordination of production and other business functions has become an integral part of the corporate information technology (IT) structure. Networking has made it possible to collect more information from the plant and to disseminate it far and wide throughout the enterprise. Geographically distributed components with lots of "intelligence" are now expected to work together. Networking has become essential for automation and is changing the way plants and factories work.

Digital Communication Networks

Many networks, such as telephone, radio, and television, are primarily analog, but the trend is definitely toward all-digital communication. So too, the networking used in automation is predominantly digital, that is, data is transmitted serially between devices as a stream of ones and zeroes. Digital communications now makes possible data transfer between devices such as transmitters, valve positioners, controllers, workstations, and servers.

More Information

A major advantage of digital communications is that a great deal of information can be communicated on a single cable. Instead of one hardwired cable for each variable, thousands and even millions of pieces of information can be communicated along just one network cable. This makes it possible to extract much more information from each device than was realistically possible using analog signals. For example, before digital communications was introduced it was impossible to remotely transmit anything other than simple I/O. Tuning and controller settings had to be done locally (figure 1-1). Therefore all controllers had to be placed in large panels lining the walls of the control room to enable operation directly from the controller faceplate. Sensors and actuators were hardwired to their controllers using an individual dedicated pair of wires and transmitting nothing more than a single process or manipulated variable. The analog signal only traveled in one direction, from the transmitter to the controller or from the controller to the positioner.

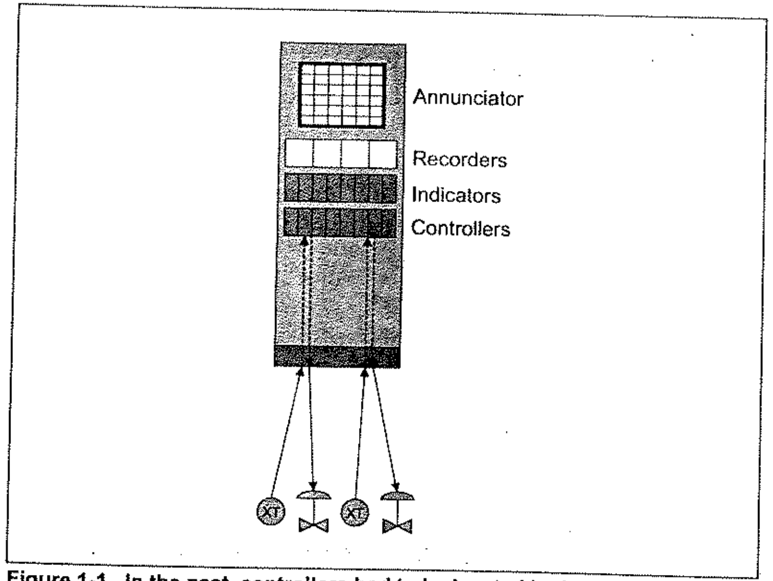


Figure 1-1. In the past, controllers had to be located in the control room panel.

The advent of digital communications made it possible for the DCS and PLC controllers to be placed away from the control room in an auxiliary rack room. All the supervisory information for hundreds of loops and monitoring points could be transmitted to the operator console in the control room over a single network. Digital communications carry not only I/O like process and manipulated variables but also operational information such as setpoint and mode, alarms, and tuning in both directions to and from the control room. Communications thus enabled distributed processing, and diagnostic, configuration, range, identification, and other information could now be added, initially in controllers but then also in field instruments such as transmitters and valve positioners. Thanks to communications, field instruments now perform not only a basic measurement or actuation but also have features and functions for control and asset management.

Multidrop

A second major benefit of digital communications is the capacity to connect several devices to the same single pair of wires to form a multidrop network that shares a common communications media (figure 1-2). Compared to running a separate wire for each device, this reduces the wiring requirement, especially for field-mounted instrumentation involving large distances and many devices. Even by putting just a few devices on each pair of wire, the amount of cable required is greatly reduced, translating into hardware and installation savings.

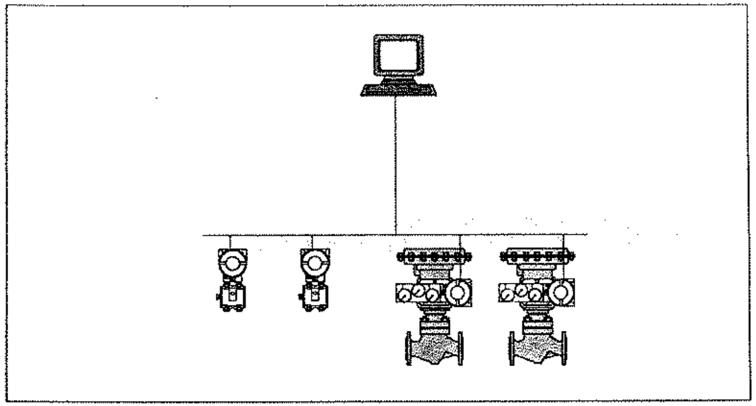


Figure 1-2. Network nodes sharing a common media.

The communicating devices on the network are called *nodes*, and each node is given a different address that distinguishes it from the other devices. This makes it possible to interrogate and send messages to any one specific device.

In the simplest form of communication, a device such as a host workstation or PLC is the master that sends requests to read or write a value to other devices such as field instruments, which are called *slaves* (figure 1-3). The slave that was addressed then responds to the request. An example of this is a HART® or PROFIBUS master configuration tool or handheld terminal writing a parameter in a slave positioner from time to time, acyclically. In networks with no specific master or slaves such as FOUNDATION™ Fieldbus this method is called "client/server": a device acting as a client requests, and the device acting as server responds. Another example of the master/slave configuration is a master PLC reading a process value from a slave transmitter and then after executing a control algorithm writing the output to a slave positioner. For PROFIBUS closed-loop control this reading and writing is repeated cyclically.

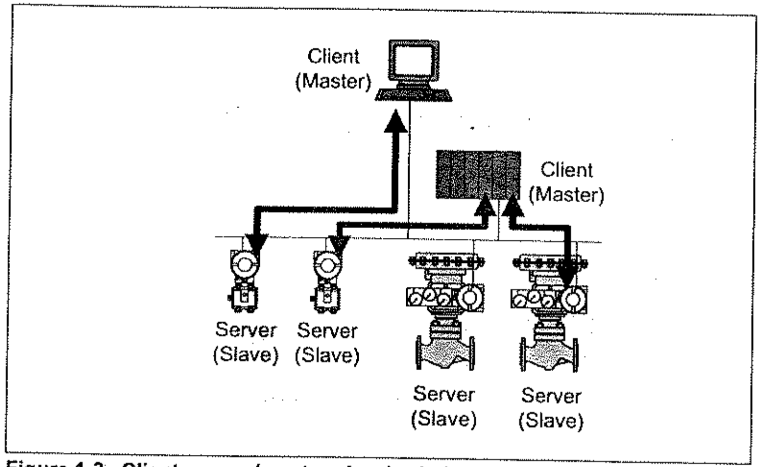


Figure 1-3. Client-server (master-slave) relationship.

Another mode of communications that is ideal for cyclic communication is where a device acting as a "publisher" broadcasts a value that is then used by all interested devices, which act as "subscribers" (figure 1-4). This is very efficient because the value is transmitted directly from one field device to another in one single communication, reaching several subscribers at once. This method is used by FOUNDATION Fieldbus for closed-loop control. Communicating from one device to another without going through a central master is called peer-to-peer communication.

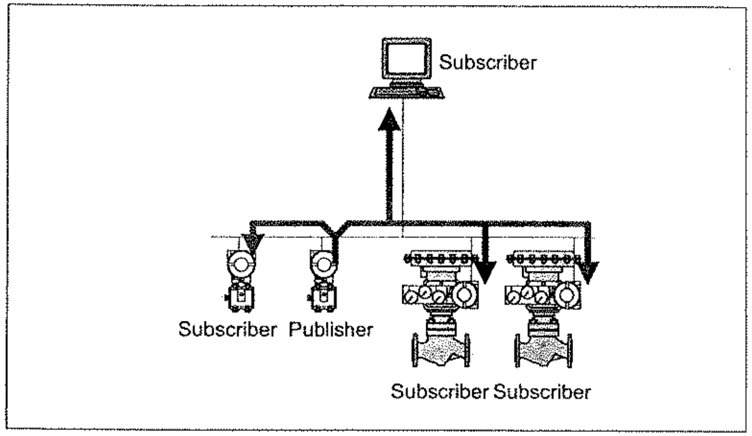


Figure 1-4. Publisher-subscriber relationship.

A third mode of communication is when a device acting as a "source" transmits a message to a device acting as a "sink" without the sink having to solicit the data (figure 1-5). While the state remains the same it is not communicated. The transmission is only made when there is a change of state sometimes called "report by exception", e.g. when an alarm occurs. This configuration is ideal for environments where operators want devices to report process alarms or fault events as they occur, while otherwise remaining silent.

Rather elaborate schemes are used by all protocols to ensure that no two devices communicate at the same time. This and other aspects of digital communications networks are explained in chapter 11.

Robust

In a 4-20 mA analog system value is transmitted by the infinite variation of a current. A signal error just changes a valid signal into another valid signal. The signal from even the most accurate analog transmitter may be totally inaccurate by the time it reaches the controller. Digital communications has the advantage of being a very robust signal with only two valid states (one and zero). It is transmitted directly or encoded in some form and is therefore less

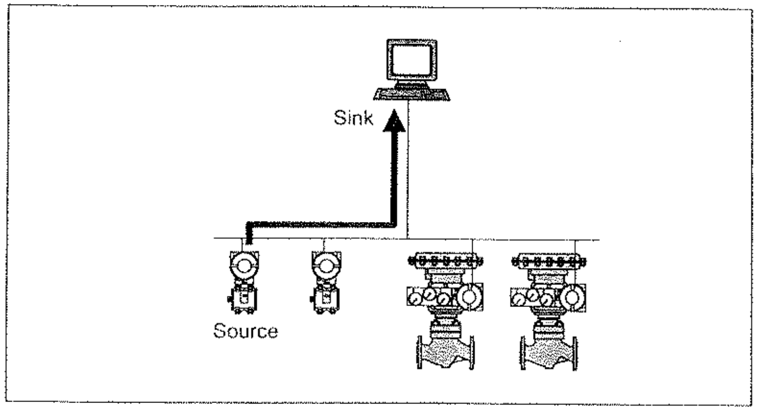


Figure 1-5. Source-sink relationship.

sensitive to distortion than an analog signal. Even more importantly, by using error-checking techniques it is possible to detect if the digital signal has been distorted, and if it has, to discard the message and possibly ask to have it retransmitted. Signal distortion cannot be detected in an analog system because a distorted signal still looks like a valid process signal. An analog signal that should be 19 mA may jump between 18.97 and 19.03 mA because of electrical interference or be limited to 18 mA because of insufficient supply voltage. There is no way to tell this, however, because it is still a valid signal. Operators may suspect a noisy or limited signal, but there is no way to tell what is distortion and what is the real process change. However, a received digital signal is true to what was originally transmitted. The superior fidelity of digital signals over analog signals is why they are used in compact disks as well as in automation; it results not only in higher accuracy but also in greater confidence level.

Interoperability

A potential problem with digital communications is that there are many different ways to do it. The method of representing, encoding, and transmitting the data is called the *protocol*. Manufacturers have devised many different protocols, and products designed for one protocol cannot work with those designed for another. One of the goals of standardization committees is to define a standard protocol that all devices can follow, thus making it possible for products from different manufacturers to interoperate, that is, work with each other. A key point is that a system's power is not defined by the capability of each of its individual devices but by the ability of these devices to communicate with each other. Two best-in-class and ever so powerful devices that don't integrate seamlessly do not create a solution as powerful as two simpler devices that use a standard protocol. For the same reason, the subsystems for basic, critical, and advanced control in a plant must also have open interfaces. Chapter 11 describes exactly how some of these protocols used in process control work, their similarities as well as their differences. It is not necessary to understand how the buses work in order to use them, however. The buses are designed such that the complexity of their function is hidden; as a result, they are easy to use.

Automation Networking Application Areas

Networking is used in all areas of automation. In factory automation, process automation and building automation networks perform diverse tasks. Likewise, there are distinct differences between tasks performed for applications in different industry sectors that all have unique characteristics and consequently varying requirements. The way devices are connected, configured, and exchange data also differ.

There is no one-size-fits-all for industrial networks; rather, buses are optimized for different characteristics. For example, factory automation and process automation are often used in harsh and hazardous environments where people, nature, and expensive machinery are at stake or where a production interruption is costly. These requirements contrast significantly with building automation, for example, where keeping costs low is a main driving force.

Factory Automation

Factories with assembly-line manufacturing, as in the automotive, bottling, and machinery industries, are predominantly controlled using discrete logic and sensors that sense whether or not, for example, a process machine has a box standing in front of it. The network types ideal for simple discrete I/O focus on low overhead and small data packets, but they are unsuitable for larger messages like configuration download and the like. Examples of this network type are Seriplex®, Interbus-S, and AS-I (AS-Interface), which are sometimes called *sensor buses* or *bit level buses*. Other more advanced protocols oriented toward discrete logic include DeviceNet™, ControlNet™, and PROFIBUS (DP and FMS application profiles). These buses are sometimes referred to as *device buses* or *byte-level buses*. Factory automation involves fast-moving machinery and therefore requires quicker response than slower processes. Traditionally, these tasks have been handled by PLCs.

Process Automation

Process plants in industry segments like refining, pulp & paper, power, and chemicals are dominated by continuous regulatory control. Measurement is analog (here meaning scalar values transmitted digitally), and actuation is modulating. Of course, process industries also use some discrete control and the predominantly discrete manufacturing industries use some discrete. Fieldbus on/off valves are already available in the market, as are small remotely mounted I/O modules for discrete sensors. In the past, a DCS or single-loop controller did this.

Process-related networks include FOUNDATION Fieldbus, PROFIBUS (PA application profile), and HART—they are the focus of this book. All these buses as a category are now typically referred to as fieldbus (without the capital *f*), though some would argue that one or the other does not belong. These three protocols were specifically designed for bus-powered field instruments with predefined parameters and commands for asset management information like identification, diagnostics, materials of construction, and functions for calibration and commissioning. In terms of size, the networks used in industrial automation are considered to constitute local area networks (LAN) spanning areas no greater than a kilometer or two in diameter and typically confined to a single building or a group of buildings. Networks that extend only a few meters are insufficient, and networks that span cities or even the globe are overkill.

Field and Host Tier Networks

Even within control systems for the process sector there is a need for different network characteristics at each tier of the control system hierarchy. At the field end there are instruments such as transmitters and valve positioners that have their specific needs, and at the host level there are workstations, linking devices, and controllers that have other needs (figure 1-6).

When fieldbus began to evolve, the process industry put a large number of requirements on the field-level network that were not met by other types of networks. Many new design considerations needed to be taken into account. On the upper tier, data from all the field-level networks have to be marshaled onto a single host-level network that also serves any tasks the plant may have that seem related to factory automation.

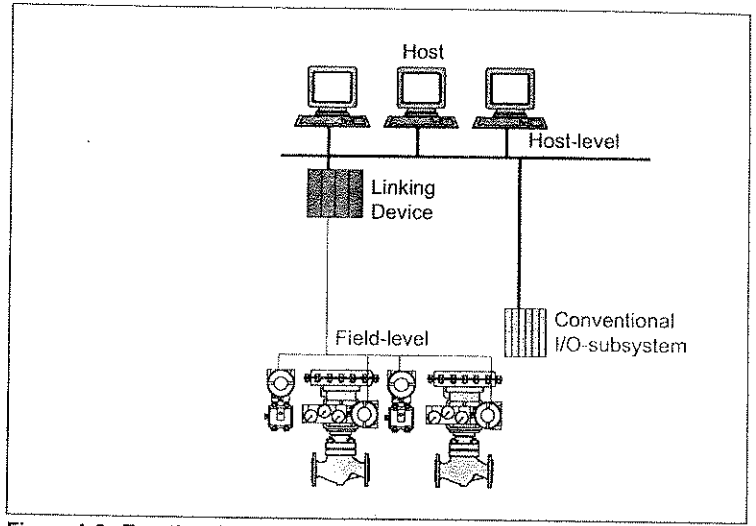


Figure 1-6. Two-tiered automation network architecture.

Field Level

At the field level, the dominant protocols for process instruments are HART, FOUNDATION Fieldbus H1, and PROFIBUS PA. HART is significantly different from the other two in that it is a so-called smart protocol, that is a combination of digital communication simultaneously superimposed on a conventional 4-20 mA signal. As such, the HART protocol has been an ideal intermediate solution in the transition from analog. HART is compatible with existing analog recorders, controllers, and indicators while at the same time it makes possible remote configuration and diagnostics using digital communication. The HART protocol does allow several devices to be multidropped on a single pair of wires, but this is a capability infrequently explored because of the low update speed, typically half a second per device. For a vast majority of installations HART devices are connected point to point, that is, one pair of wires for each device and a handheld connected temporarily from time to time for configuration and maintenance. Both FOUNDATION Fieldbus H1 and PROFIBUS PA are completely digital and even use identical wiring, following the IEC 61158-2 standard. However, beyond that there are major differences between these two protocols, and depending on the desired system architecture one may be more suitable than the other.

At the field level, instruments appear in large quantities, often in the hundreds or thousands. The wire runs are very long, as the network cable must run from the control room all the way into the field, up towers, and then branching out to devices scattered throughout the site. Because there is a limit to the number of devices that can be multidropped on each network, even a medium-sized plant may have many network cables running into the field, although substantially fewer than if point-to-point wiring was used. The field-level networks were therefore designed to enable very long wire runs and to allow field devices to take their power from the network. Only a single pair of wires carries both the device's power and the digital communications signal. This eliminates the need for a separate power cable, thus keeping the wiring simple and inexpensive.

As another measure to keep costs down, designers chose a moderate field-level network speed so normal instrument-grade cable could be used instead of special data cable. No special connectors, couplers, or hubs are required either, which makes it possible to use rugged and weatherproof connections. The grade of cable used for conventional instrument connections on most sites is more than sufficient for fieldbus networking. As a result, it is possible to reuse that cable when an existing plant is migrated to fieldbus. In hazardous process environments where flammable fluids are present intrinsic safety is many times the preferred protection method. The field-level networks were therefore designed to allow safety barriers to be installed on the bus.

Because designers chose a moderate field-level network speed the devices connected to it do not require a great deal of CPU processing power to handle the communication quickly. As a result, they also consume very little power. Because the low power consumption results in low voltage drop along the wire, it is therefore possible to multidrop several devices on the network even for long wire distances and even when using intrinsic safety barriers. Another great advantage of field-level networks is that they provide a lot of freedom when it comes to network topology since wires can be run quite freely. Finally, these fieldbus networks were also designed to operate in the often rather harsh, electrically noisy environment found on site.

Host Level

At the host level, the Ethernet network standard is already the dominant wiring technology (figure 1-7). There are many protocols built on Ethernet wiring, including FOUNDATION Fieldbus HSE, PROFInet, Modbus/TCP, and the like. Sites employing fieldbus instrumentation and asset management software can expect to

encounter a steep rise in bandwidth requirements and must therefore have a high-speed network at the host level.

The field-level networks have made it possible to retrieve so much more data from the field instruments that an information explosion has resulted, one that old proprietary control level networks are unable to cope with. Ethernet provides the throughput required to transfer the large amount of data used for traditional plant operation and historical trending; for new capabilities for remote diagnostics, maintenance, and configuration; and for the quick response necessary for factory automation task. Ethernet was chosen for these applications because its high speed enables it to carry all this information. Moreover, Ethernet is already a standard and consequently is well understood and widely used. A large variety of equipment and solutions for Ethernet is available. Ethernet wiring is discussed in chapter 3, "Installation and Commissioning."

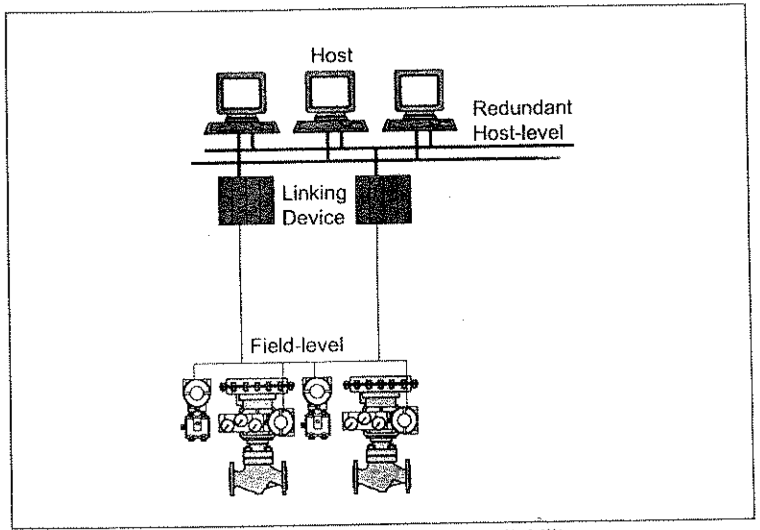


Figure 1-7. Host-level network redundancy for availability.

In many applications, one of the key requirements for the host-level protocol is availability. The network must be fault tolerant—up and running even in the presence of a fault. This is extremely critical at the host level since the entire site is operated and supervised over this network. Downtime can be very disruptive and cause heavy losses; a complete breakdown of the network would be extremely serious. Though Ethernet originated in the office environment, rugged industrial-grade (as opposed to commercial-grade) accessories and wiring schemes can be used. The host-level network was designed so redundancy may be used, making the network fault tolerant. Industrial-grade networks that use several layers of redundancy and industrial-hardened components can handle many simultaneous faults.

Physical remoteness is less important for the host-level network because it is typically confined within the control room, and the distance Ethernet provides is therefore sufficient. An advantage of an established standard like Ethernet is that several media options are available. On copper wire Ethernet is unsuitable for the field because it does not run long distances. It is therefore limited to use within the control room (i.e., a "hostbus" rather than a fieldbus). However, optical fiber Ethernet can run very long distances, as can radio signals, making Ethernet suitable for remote applications.

Table 1-1. Comparison of field and host level characteristics

	Field Level	Host Level
Speed	Low	High
Distance	Long	Short*
Two-wire	Yes	No
Multidrop	Yes	No*
Bus power	Yes	No
Intrinsically safe	Yes	No
Media redundancy	No	Yes
Note: Fiber-optic Ethernets can run long distances. Using the old coaxial cable wiring Ethernet can also be multidrop, but this introduces other problems.		

The host-level network ties together all the subsystems the process automation system might have. In addition to the basic control function, a plant often has package units for auxiliary functions such as boilers or compressors that are bought ready-made. They have their own controls that need to be integrated with the rest of the system (figure 1-8). For example, a refinery may have a safety shutdown system, a paper mill may have a web scanner, and a chemical plant may have an advanced control system. Subsystems based on a standard protocol on Ethernet can simply be plugged into the rest of the system.

The host-level network tier makes large systems possible by linking together field-level networks from different areas around the site. Intra-area control and supervision becomes possible. The host-

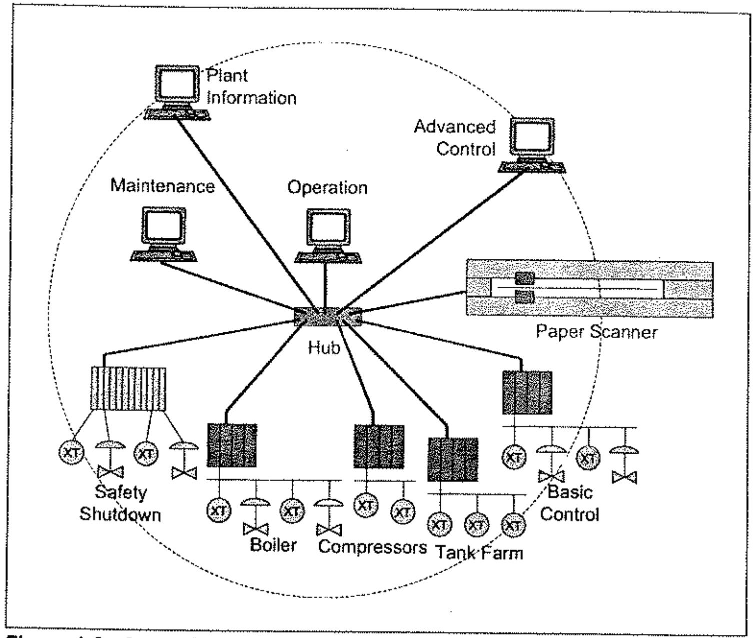


Figure 1-8. A standard host- level network ties distributed subsystems together.

level protocol is also the link to business systems, either directly or via historians and other plant information software.

It is important to remember that the Ethernet standard is not a complete protocol. Essentially, Ethernet only specifies different options for cables and how devices on the network access the bus. Ethernet does not specify data formats or the semantics of the data. Even when used with other technologies like TCP/IP and UDP the protocol is incomplete. Several control system manufacturers have been using Ethernet for many years, but each one has implemented it with data formats and functionality different from the others. Even with TCP/IP, most of the Ethernet networks used in control systems on the market today are in fact proprietary since other devices cannot access and interpret the information even though connected on the same wire and existing without conflict. As a result, take great care when buying products and systems for Ethernet; they are often not as they appear to be. TCP, UDP, and IP are discussed in chapter 4, configuration.

It is a good idea to look for complete open protocols based on Ethernet so devices and subsystems from different sources can talk to each other, even peer to peer.

Homogeneous Network Architecture

Because of their almost opposite requirements, different network features are required at the field and host levels. Because the field-level network is slow it is unsuitable for the host level, and because the host level has too limited a distance it is unlikely it will be seen in the field. The field-level network takes the place of the traditional protocols for smart instruments and I/O subsystems, and the host-level network takes the place of the control network and business network. The host-level network in the control system uses the same networking technology as the business network so they can be integrated seamlessly. A simple router between the networks safeguards performance by keeping pure business communication traffic separate from pure control communication traffic.

For easy and tight system integration it is important to select a homogeneous network architecture in which the protocols at the higher and lower tiers are essentially the same but just traveling on different media. This will ensure transparency and a minimum of problems with communication mapping and interoperability. Fortunately, there are protocols available in such "suites." Good combinations would be FOUNDATION Fieldbus H1 and HSE or PROFIBUS PA and PROFInet. If a proprietary protocol is used at the host level or somewhere in the link between the instruments and the operator important functionality and interoperability may be lost. This may force engineers to perform time-consuming mapping of parameters between protocols.

The use of the same technology throughout the system greatly simplifies the initial engineering and deployment of the system as well as its ongoing operation and management. Engineers can readily work with different parts of the system without retraining.

History of Fieldbus

The history of process control networks is very much the history of the IEC 61158 Fieldbus standard.

Lack of Interoperability

When digital communications first began to appear every vendor invented its own protocol independently of others. Soon many different proprietary protocols were in the market, and products could only work with other products from the same vendor. Moreover, documentation on the operation of these protocols was typically not available, and the technology was generally protected by patents. Other manufacturers would have to pay high licensing fees to implement the technology in their products—if they were allowed to do so at all.

This situation resulted in several disadvantages. One was that no vendor had a range of products wide enough to provide all the parts a site required. The selection of equipment was very limited, so it was always necessary to mix and match equipment from different suppliers. Moreover, one supplier is never the best at everything, so it was desirable to buy the device types from the manufacturers that were specialists in each particular area. Because the equipment from different suppliers had incompatible protocols a site was stuck with a few undesirable options: either choosing the preferred device despite its poor integratibility with the rest or settle for the less-than-best device to gain better integration. Most of the time, however, it was not possible to network the parts together, resulting in isolated islands of automation.

In one common scenario, a PLC and a DCS would have to be connected, but digital integration of the system was impossible since each component communicated using a different protocol. If the manufacturers allowed licensing and provided proper documentation, a communication driver could be developed—but at great expense in time and money. A third party often developed the drivers, and when communication problems arose the parties would point fingers at each other. To complicate matters further, one driver was required for every combination of hardware and software, producing an unmanageable situation for suppliers too. Many times no communication at all was possible, and to pass data subsystems had to fall back on conventional analog and discrete signals. Because of the protocol differences third-party field instruments could not be integrated with the DCS to fully benefit from their intelligence, nor could one supplier's handheld terminal or other configuration tool work with a device from another.

The Need for a Standard

Once a proprietary system had been purchased the plant was essentially "locked in" by the manufacturer. To maintain system integration the plant would have to purchase replacement transmitters from the system supplier, who was also the only one that could do system expansions. Because the system supplier at this point no longer had any competition, replacement parts and extras would be much costlier than they were for the first system. Many plants were aware of this but were still willing to pay the price of being tied to a single manufacturer simply because of the high cost of struggling with system integration in a situation where incompatible protocols required drivers.

Being locked in can be dangerously costly, so many governments prevent the use of proprietary technologies in public projects. Many instrument suppliers were also displeased with the situation. Despite the fact that they often had higher-performance products, the instrument suppliers were unable to compete with the systems suppliers simply because of the incompatibilities. Furthermore, adapting their products to a myriad of protocols was extremely costly, driving up product prices. As is often the case when standards are lacking, there was anarchy in the market.

Standardization

Because the situation was clearly intolerable, in 1985 industry experts began work on a vendor-independent fieldbus standard. Networking is a key element of an open system, and it was paramount that an interoperable fieldbus be developed that was supported by multiple vendors and based on a freely available standard without licensing. Standardization is an enormous task that not only involves the development of a technology but has economic and political implications for factories, manufacturers, and even nations.

Because of the unique needs of the process control environment no existing standard for networking could be used. A new technology had to be developed for the standard that provided bus power, intrinsic safety, the ability to communicate long distances over normal instrument wires, and so on. This development process led to an international fieldbus that could not move as fast as other networks that used an existing platform from telecommunications or automotive industry. Nevertheless, it filled an important need. Many of the systems suppliers who participated in developing the standard had vested interests in the old technology and a comfortable market share in the proprietary paradigm. Standards allow competitors to take away customers that had previously been locked in to a particular supplier's proprietary technology. Thus, these proprietary suppliers had a responsibility to their shareholders to see the fieldbus standard fail so they could avoid tougher competition.

Naturally, some companies and nations wanted to see their existing technology and national standards adopted as the international fieldbus.

These factors further contributed to the delay in the ratification of the single fieldbus standard. The world failed to agree on a single standard protocol, and as a result several competing and noncompatible bus technologies are now being included in a multi-part standard that is not yet fully completed. Parts of both FOUNDATION Fieldbus and PROFIBUS, though not HART, are elements of this standard, but devices of these two types cannot communicate with each other since the protocols are not compatible.

Industry Groups

Frustrated with the delays in the development of standards, manufacturers and end users formed organizations to fast-track the creation of open fieldbus specifications. In 1992, the Interoperable Systems Project (ISP) was formed to develop a technology partly based on PROFIBUS and soon thereafter WorldFIP to develop another based on FIP. Because these are open organizations that develop and maintain the technology, both projects have the openness of a true international standard. The organizations split and merged, but for the process industries organizations had by 1994 essentially crystallized into the Fieldbus Foundation and Profibus International.

FOUNDATION Fieldbus and PROFIBUS PA have a common heritage in the ISP technology; therefore, the concept of block, parameter, mode, and status is very similar. The FOUNDATION H1 technology was released soon after followed by PROFIBUS PA in 1996. FOUNDATION HSE was released in 2000, and by 2001 PROFInet was already on its way. For several years now, manufacturers have been delivering products based on these specifications, and plants are already reaping their benefits. Large parts of the specifications are being adopted as national standards and will soon also become an international standard. However, some parts of both technologies are still under the control of the organizations.

During fieldbus's long gestation period some originally proprietary protocols such as HART and Modbus were opened up and made available to other manufacturers. These were tremendously successful in filling the gap and now have an enormous installed base and will keep selling for years to come.

Advantages of Standards

Once the standards were in place plants could truly begin to benefit from integration without paying the high price of being tied to a single manufacturer (figure 1-9). Standards have already resulted in compatible equipment now available from several suppliers. More than one company now manufactures device types that are based on the same fieldbus technology. This has led to a competitive open market, a desirable development because it reduces prices. Sites that employ standards are protected from proprietary solutions that force them to be dependent on a single vendor. Similarly, the plants that have adopted standards have many more options available for devices and software. This enables them to find solutions for their very diverse application needs, needs that cannot be met by a single supplier but require equipment from several manufacturers. Device manufactures can once again concentrate on true innovations rather than tweaking communication protocols.

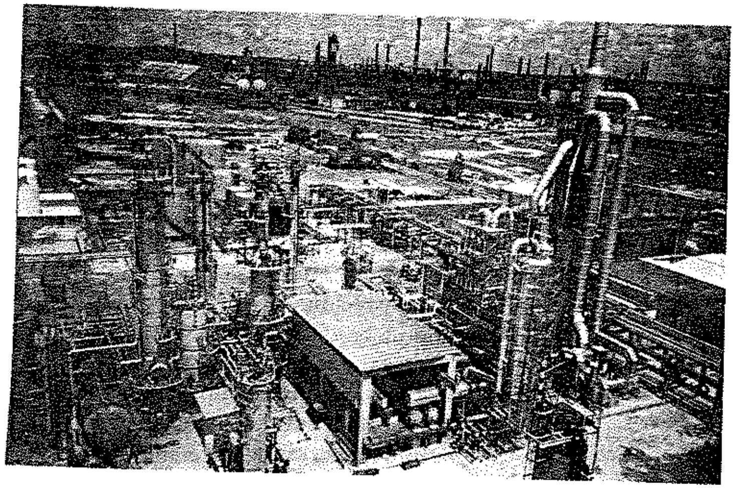


Figure 1-9. The world's first H1 fieldbus plant. (Courtesy of Smar)

Evolution of Control System Architecture

Field signaling and system architecture developed in very close-knit fashion. Every improvement in signal transmission has subsequently led to an increased level of system decentralization and better access to field information. In the pneumatic era the controller was typically situated in the field and there operated locally. There was therefore no system to speak of. With the analog current loop it became easier to bring a signal from the transmitters in the field to a central controller in the control room and then from there back out to the valves again. In the completely centralized direct digital control (DDC) architecture the complete control strategy was executed in a computer. Because all the functions were concentrated into a computer the entire system with all of its loops would fail if there were even a single fault. For this reason, it was not uncommon to have local pneumatic controllers existing in the field on standby, ready to be put in operation once the DDC failed. Clearly, the centralized architecture had some serious availability issues, which led in the early 1970s to the introduction of more decentralized programmable logic controller (PLC) and distributed control system (DCS) architecture.

DCS and PLC Architecture

The DCS and PLC emerged with the advent of digital communication, but these architectures were also designed based on 4-20 mA for field transmitters and valve positioners. However, the DCS was a great improvement over the DDC in that the controls were now distributed over several smaller controllers that shared the tasks, each one handling perhaps thirty control loops. This had the immediate benefit that a single fault would only affect part of the plant, not all of it as with the DDC. In other words, a higher level of distribution increased the availability of the system.

A secondary benefit was that the configuration could be better structured where separate plant units were also kept separate in configuration and controllers. The DCS and PLC architectures are characterized by conventional I/O (input/output) subsystems or "nests" in which racks of I/O modules are networked to their respective centralized controller via an I/O-subsystem network. Field instruments were predominantly conventional analog devices. The controllers are networked with each other and to the workstations via a control-level network. There may also be a plant-level network at the very top that links the workstations to the business environment. The DCS evolved over many years, and such capabilities as communications interfaces for smart instruments that used the manufacturer's proprietary protocol became an option. This allowed some degree of configuration and check. Not all of the smart instrument protocols allowed simultaneous 4-20 mA and communication. For this reason, many were unable to use the communication feature. However, most DCS models did not provide HART interface because all the system manufacturers had their own competing proprietary protocols. Thus, plants were inclined to buy the field instruments from the system supplier rather than from third parties.

A DCS can often have, in all, as many as four different tiers of networking, each with a different technology: device, I/O subsystem, controllers, and plant-wide integration to business applications (figure 1-10). All these levels of hardware and networking result in a rather complex and costly system.

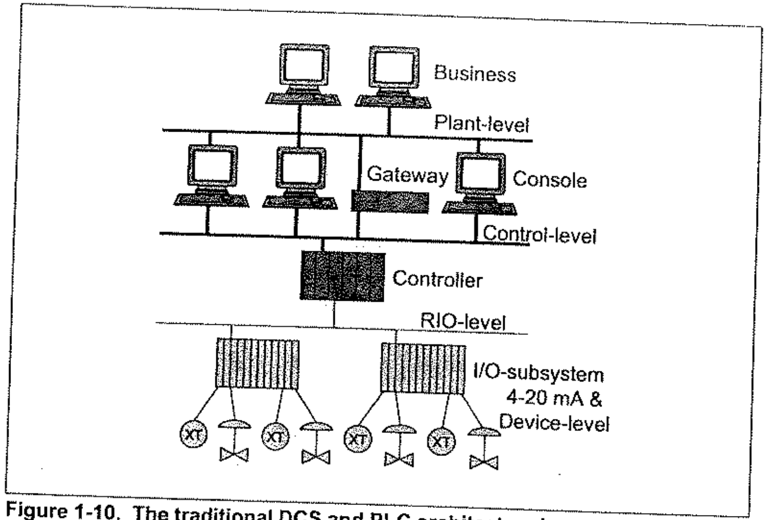


Figure 1-10. The traditional DCS and PLC architecture has multiple network levels.

When introduced, the DCS was christened "distributed" because it was less centralized than the DDC architecture. By today's standards, however, the DCS is considered centralized. This architecture is relatively vulnerable because just one failure may have widespread consequences. Because of this vulnerability, redundancy of controllers, I/O-subsystem networking, I/O modules, and the like is a must to avoid a total loss of control. Of course, redundancy at every level means complexity and high price.

FCS Architecture

The FOUNDATION Fieldbus specification is uniquely different from other networking technologies in that it is not only a communications protocol but also a programming language for building control strategies. One of the possibilities that a standard programming language and powerful communications features enable is the ability to perform control that is distributed into the field devices rather than a central controller. For example, it is common for the valve positioner to act as a controller for the loop it is part of. It executes the PID function block but only for its own loop, not for other loops. This new architecture based on field device capability is called Field Control System (FCS) and is an alternative to DCS (figure 1-11) in that the architecture is not controller-centric. It does not treat every field device as a peripheral. Because of its decentralized nature the FCS architecture has advantages like high availability, greater scalability, and lower cost. The FCS architecture has evolved from the concept of the DCS carrying the original concept further, and the result is a system that is more distributed and therefore less vulnerable to faults.

In the FCS architecture the instruments on the field-level networks are connected to the workstations via a linking device to the host-level network. Thus, there are only two network tiers in a FCS. Typically, the field instruments perform the regulatory control that in the process industries accounts for the bulk of the automation tasks. The linking device or a central controller may perform discrete logic and sequence controls. When control is performed in the field devices the number of central controllers that is required is drastically reduced and in some cases eliminated altogether. This dramatically cuts the cost of the system. In other words, wire savings are not the only hardware savings that can be achieved by using bus technology. Since the central controllers have the computation-intensive regulatory controls offloaded they are freed up to execute other controls with higher performance, thus improving controls.

Because in the FCS no one controller handles multiple loops the problem of a single fault affecting a large part of the plant is largely eliminated. However, even in an FCS a centralized controller can often be found handling discrete I/O and controls since these functions are still seldom networked. Whenever a plant uses centralized controllers, it should employ redundancy if availability is a necessity.

It may at first be hard to comprehend how small field device controllers could replace a "unit controller" to control a large plant. The secret behind this concept is that each device handles only one loop. By networking hundreds or thousands of devices together the combined power of the microprocessors exceeds that found in earlier systems. The control task is broken up into its components

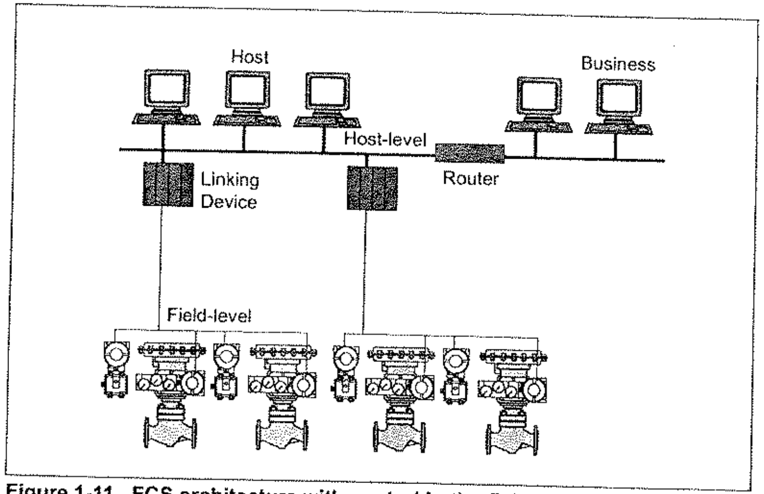


Figure 1-11. FCS architecture with control in the field devices.

and distributed among the field devices working in parallel, with each device responsible for its loop. Since these devices work simultaneously a true multitasking system is achieved, something that cannot be realized using only a single processor. The net result is therefore very good performance, and the more devices that are added the more powerful the system becomes. This increased power has made it possible to eliminate the need to scale analog values. For centralized systems, this scaling had not always been possible because it loaded the processor too much. Floating-point format is now used throughout the control strategy.

Host versus System

Because a 4-20 mA signal carries only a single piece of information and only in one direction, operators had no way of determining what was going on within analog field devices. It was impossible to perform configuration, diagnostics, and other checks from the system console. In the cases where smart instruments had been adopted a handheld terminal was usually used to extract any additional information. Conventional and even smart devices were not integrated within the control system. The operator's view extended down to the controllers and possibly to the I/O subsystem, but no further. Because the field instruments were isolated entities, they were treated as separate from the control system rather than part of it.

In an FCS the field instruments are an integral part of the system as a whole. All that remains of what used to be called the system is the workstations and linking devices. The workstations that connect directly to the host-level network are simply referred to as the host (figure 1-12).

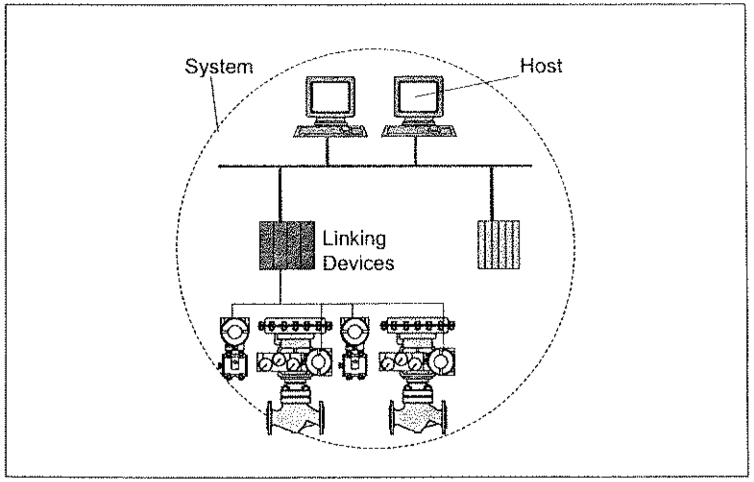


Figure 1-12. Field devices, and host, are integral parts of the system.

Basic Network Differences

It is technically possible to use FOUNDATION Fieldbus or PROFIBUS PA technology in any kind of system architecture. Systems based on conventional architecture can also benefit from the wire reduction made possible by field-level networks. However, few traditional systems have native support for fieldbus.

Communications Subsystem Differences

The communications interfaces required by a host are different for the pure digital communication in FOUNDATION Fieldbus and PROFIBUS on the one hand and for the hybrid of analog and digital for HART on the other. For PROFIBUS and FOUNDATION Fieldbus a single integrated network architecture is used for I/O as well as for asset management. Because its communication speed is low HART relies on the analog 4-20 mA signal for real-time process I/O and a HART device is therefore connected point to point. In most systems, the 4-20 mA only connects to conventional I/O modules via individual wires, and any communication with the device is performed with a temporarily connected portable handheld terminal. However, plants should enhance HART installations by integrating a permanently fixed communications subsystem that is connected in parallel with conventional I/O. This brings full field device data into the control room, making it easy to benefit from device intelligence. A HART multiplexer is connected to all smart devices, giving the device configuration tool complete access (figure 1-13). Alternatively, an I/O subsystem with built-in HART capability may be used. Without digital integration many plant operation improvements are simply not possible.

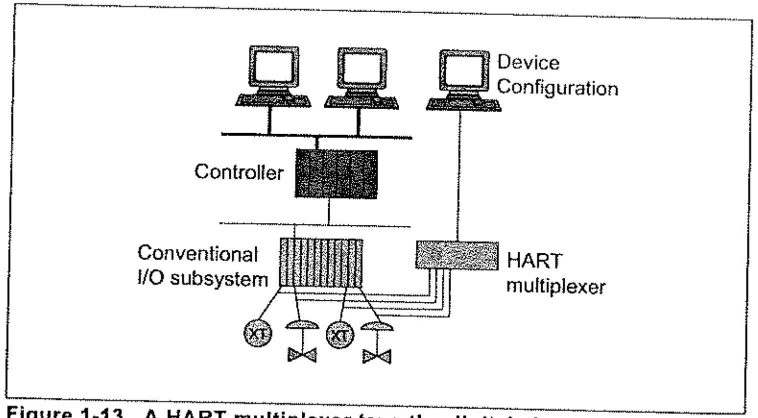


Figure 1-13. A HART multiplexer taps the digital signal from smart instruments.

Because HART blends the benefit of digital communications with complete analog compatibility the transition from pure 4-20 mA to HART became easy and HART's success was assured. In a pure digital system based on FOUNDATION Fieldbus or PROFIBUS, full information access comes built in since all communication is digital, the networking infrastructure is in place for I/O, and no additional hardware or wiring is required.

Other Technical Differences

The HART and PROFIBUS technologies do not have a control strategy programming language. FOUNDATION Fieldbus has a standard function block language and publisher/subscriber communication. It therefore has the ability to constitute an FCS, but it can also be used in a DCS or PLC. HART and PROFIBUS are only used in DCS or PLC architectures, be it a traditional embedded PLC or PC-based software logic. A traditional DCS using FOUNDATION Fieldbus would not achieve the controller reduction and network simplification savings achieved by the FCS. Thus, one of the main criteria for selecting a bus technology is what architecture is desired. For a DCS or PLC, either one can be used; for a full-fledged FCS only FOUNDATION Fieldbus is possible. FOUNDATION Fieldbus has a number of useful communication features not offered by most other protocols. These include automatic device detection and address assignment for Plug-and-Play installation and time synchronization. PROFIBUS has PROFIsafe for communication between instruments in safety- related systems, which FOUNDATION and HART do not offer.

Commercial Differences

Of course, there are other criteria to consider when selecting the principal network to be used in the system. Are the device types and tools the plant requires available in a version that has the desired protocol? Are there multiple vendors of the product types the plant requires so as to ensure a competitive price now and in the future? Do the manufacturers of the products that will be used have good local support through either their own offices or representatives, or do the products have to be imported without support?

EXERCISES

1.1 Is all networking digital?

1.2 Is HART a master/slave protocol?

1.3 Is FOUNDATION Fieldbus also a control strategy programming language?

1.4 Is publisher/subscriber a more efficient way of communicating cyclic data than master/slave (client/server)?

1.5 Which type of automation generally requires faster network response times, factory automation or process automation?

1.6 Is Ethernet a protocol?

1.7 Does a distributed architecture increase availability?

BIBLIOGRAPHY

1. *Application Guide, HCF LIT 34*, HART Communication Foundation, Austin, TX, 1999.

2. *Draft PROFIBUS Profile, Fail Safe with PROFIBUS*, Revision 1.0, Order No. 3.092, April 1999.
3. *Technical Description, Order Number 4.002*, PROFIBUS Nutzenorganisation e.V., September 1999.
4. *Technical Overview, FD-043*, Revision 2.0, Fieldbus Foundation, 1998.

2

Benefits, Savings, and Doubts

When embarking on a project for a new plant, an expansion, or a re-instrumentation everyone on the project team must understand the benefits of fieldbus technology and how it can be used to change the way the plant is run. There are many stakeholders in the project team who will have to agree on fieldbus. Since many people in the plant's management will be hesitant to change the technology used in the plant's operation they will have to be convinced that fieldbus is the right direction for the plant. Investing in old, tried-and-tested technology carries the risk of obsolescence and will not give a site any advantage over the competition. Among the points to consider are the challenges that will be encountered when using a technology like fieldbus for the first time as well as the benefits and subsequent savings that technology can bring. Any doubts about the technology must be addressed. To win approval from management for the project, it may help to refer to these points to explain the business advantages to them. Of course, the benefits that can be achieved depend on which fieldbus technology you choose because each technology has different features. Ultimately, plants want to produce with higher quality, better profit margins, and less pollution than their competitors. Fieldbus technology enables solutions that help plants achieve these goals.

Realizing Fieldbus Benefits

Fieldbus technology should not be thought of as a "digital 4-20 mA" because it is not only I/O and it affects more than the field wiring. In other words, fieldbus does not just take the place of conventional I/O signal transmission. The advent of fieldbus technology has made possible a wide range of new capabilities throughout every level of the control system that had not previously been possible or fully explored. This includes more control in the field instruments, the measurement and transmission of multiple variables, and field device maintenance management based on self-checks etc., among other benefits. These new capabilities bring benefits that in turn translate into savings at various stages in the system life cycle, as early as the time of deployment, during operation, or even at future decommissioning. Substantial savings are achieved using capabilities that traditional systems do not have.

Interoperability

Interoperability is the ability of a device to work together with other devices. This enables easy and tighter integration of devices from different manufacturers than what has previously been possible and to some extent the substitution of a device by that of another manufacturer that also could not be done when protocols were proprietary. Interoperability is made possible by the standards and specifications that have been laid down and tests to certify that the standards are being followed.

Standard User Layer

A complete communications protocol performs many functions that have been categorized in groups or layers, as detailed in chapter 11, "How Fieldbus Works." The most visible parts are the physical layer that covers the network media on which bits get from A to B, that is, the wiring, and the user layer that defines the semantics of the data, that is, what the bits and bytes mean. To be connected to the same cable the devices need to conform to the same physical layer standard and to fully interoperate the devices need to conform to the same user layer standard. RS232, RS485, CAN (Controller Area Network), and Ethernet are well-known standards but basically only for the physical layer. None of them is a complete protocol. They essentially just specify electrical signal levels and the types of cable that can be used. These physical media do not provide any guarantee of interoperability, but they are the basis for many protocols. Though many devices and systems on the market use Ethernet, they are essentially proprietary because they do not use a standard user layer. As a result, they force the plant to develop costly drivers and configurations to enable the proper display and action on the data. Manufacturers may not even make the necessary documentation for driver development available to another supplier. Bits and bytes may be transmitted successfully from one device to another, but without a user layer standard the meaning is not understood (figure 2-1).

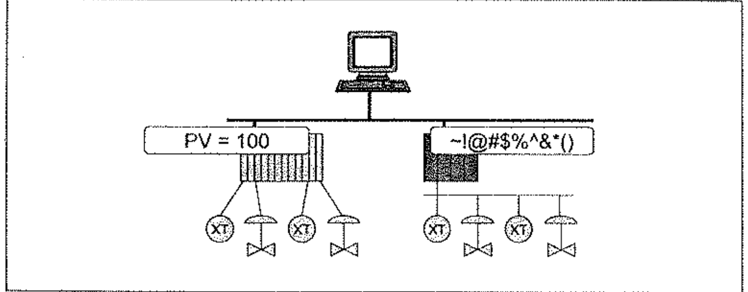


Figure 2-1. A physical layer standard does not ensure that the meaning of the data is understood.

While most communications networks are only "data pipes" pushing information from one point to another HART, FOUNDATION Fieldbus and PROFIBUS PA have gone further than most protocols. They specify a user layer with parameters for input, output, alarms, tuning, diagnostics, and configuration and also indicate how the values are to be interpreted. Using these full-fledged protocols, software and devices have a common understanding of the meaning of the bits and bytes and therefore a far higher degree of interoperability than could be achieved by, for example, Ethernet on its own. A seamless system can therefore easily be put together without patching by using drivers and data mapping or by falling back on analog signals, as is the case with proprietary systems. Physical layer standards eliminate the need for special interfaces, and user layer standards eliminate the need for custom device drivers. Using fieldbus technology it is possible to mix and match the best instruments and software applications with a minimum of worry about incompatibilities between manufacturers.

Device Description

Hosts must access complex data from devices. FOUNDATION Fieldbus and HART utilize Device Description (DD). DD is a set of files containing information about the information in the device and is provided by the manufacturer of each particular device. DD works like a kind of Rosetta stone in that it allows the host to interpret any data and to work with all functions (figure 2-2).

Using the DD the host can decode and display all of the field device information so it becomes useful to the operator. The user

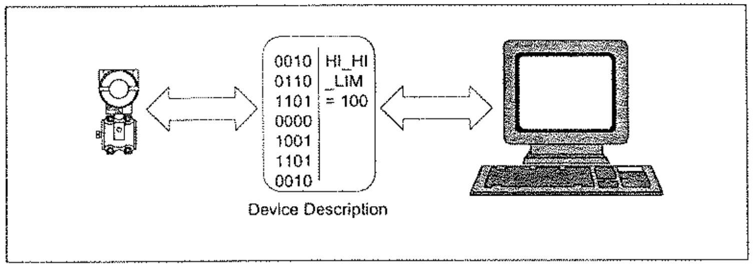


Figure 2-2. Device description information ensures that all data can be displayed.

can freely select any kind or brand of field instrument, including newer and older versions of the same device type. All the user has to do is copy the DD files onto the host computer. Manufacturers can therefore supply devices that include special parameters or unique features, and yet because of DD the access is assured regardless of host used. In other words, DD allows manufacturers to extend the usable device capability far beyond basic functions. DD can be seen as a standardized mechanism for replacing proprietary driver schemes. Well-defined standards ensure that FOUNDATION field devices are interoperable and can exchange simple I/O data with each other directly without the need for drivers.

Subsystem Interoperability

Just as the field-level network standard gives sites the freedom to select field instruments, the open host-level network lets sites mix and match subsystems from different suppliers. Until now, integrating such auxiliary systems from various manufacturers with the basic controls has been difficult because of the proprietary protocols used by all parties. Because of limited driver availability certain combinations of system parts were not possible. Using standard host-level protocol subsystems such as paper web scanners, gas chromatographs, compressor and turbine controls, and burner management will be integrated as one. It is even possible for subsystems to use different field-level networks but be integrated with each other through the same host-level protocol.

Programming Language

The FOUNDATION Fieldbus is the only bus technology to include a control programming language for building control strategies. This programming language is graphical, not textual, and uses the familiar function block diagram. A control strategy is built by selecting function blocks and linking them together so process and manipulated variables can be passed from one block to the next (figure 2-3). A library of function blocks is defined in the specification, and it includes Analog Input (AI), PID control, and Analog Output (AO), among others. Chapter 4, "Configuration," provides details about the function block configuration.

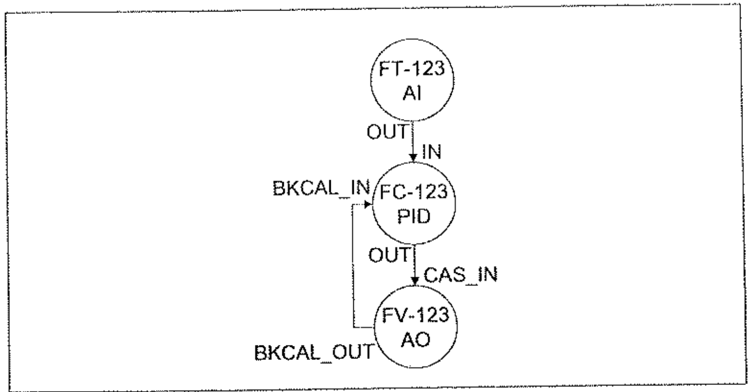


Figure 2-3. Basic PID control loop represented as a function block diagram.

Using a standard programming language offers several advantages. Equipment from different suppliers are configured in the same basic way, making it easier for engineers to work with equipment from different vendors, with little or no retraining. Training received on one system can be applied to another manufacturer's system too. Standard programming language also makes it very easy and completely transparent to execute the control strategy by mixing controlling devices from different manufacturers. Most of the time, a function block in one device is identical to that in another, and blocks always operate in the same basic way. The standard programming language makes possible great interoperability between devices since the parameters for communication between field devices have been tightly standardized. Such parameters now allow function blocks in the field devices to use the data from devices made by other manufacturers directly without further processing or interpretation. The status communication between devices is not only for display; the receiving devices take appropriate action for reset windup protection, cascade initialization, shutdown, and so on, resulting in better control and response to abnormal conditions. Reset windup protection in the PID is assured through feedback even if the valve is operated by hand. The additional benefit is a bundle of functionality with a minimum amount of engineering. Another great advantage of a standard programming language is that the execution of control strategy is synchronized with the communication, even between devices from different suppliers. This is possible because the communications protocol and programming language are defined in one and the same specification.

Device Families and Transducer Blocks

There is device information that is unique to the measurement of a particular physical property or a method of actuation and is simply is not applicable to another. Often these parameters are related to a certain measurement technology used by more than one manufacturer. To this end, the PROFIBUS PA and FOUNDATION Fieldbus devices have subsets of parameters for information that is applicable to a particular device type. For example, there is information for pressure, level, temperature and flow transmitters, valve positioners and analyzers, and so on. These sets of parameters are grouped in transducer blocks that are tailored for the pressure, flow, level and temperature measurement, valve positioning and so on and contain information applicable to each particular device type. This includes, for example, temperature sensor type selection, pressure sensor limit, calibration records and diagnostics. These transducer blocks are where the calibration and set-up of devices is performed. Similarly, HART defines a device family where each family has an associated set of commands for accessing information relevant to that particular device type.

Predefined parameter sets contain all the common information and provide a fair amount of information access and interoperability. Additional parameters supported by the device, beyond what is specified in the standard HART commands and FOUNDATION transducer blocks, are described in the DD for the respective device. This ensures that no data is proprietary.

Test and Certification

For standards to be effective they have to be followed. If devices didn't fully conform to the specification and manufacturers started doing their own thing interoperability problems would follow. The field devices based on these technologies are therefore thoroughly tested by experts independent of the manufacturers or by the manufacturers themselves using specially developed tools to guarantee that the devices are in fact interoperable. The testing includes electrical aspects as well as communication and device support file aspects. This ensures the quality of the device communication and catches any implementation errors so they are rectified before the product is sold. The rigorous testing ensures that devices should not have communication problems. A minimum of integration problems will therefore be experienced once the device has been put into operation.

Greater System Functionality

Perhaps the most important benefit of digital communications is that a virtually unlimited number of parameters can be accessed from a device. This allows multiple measurements and a wide range of features to be crammed into advanced device firmware and accessed remotely by sophisticated software. The fieldbus technologies therefore make possible a level of system functionality not previously found in field devices and hosts. Fieldbus instruments have capabilities far exceeding simple measurement and actuation. Innovative products and solutions have become available for those plants that have invested in these technologies. These standards boost innovation as manufacturers are able to make devices and software do much more. They no longer need to waste time reinventing the wheel but can take development to the next level. The biggest change brought about by fieldbus technologies is not in the devices themselves but in a new class of software functions in the host. Today, third parties are developing innovative accessories and software for devices they themselves don't manufacture.

Multiple Variable I/O

In the past, a two-wire transmitter had a 4-20 mA output that could only transmit a single piece of information—one measurement value. Though two-wire multivariable transmitters have been around for several years they never gained mainstream acceptance because the digital communication was slow and as a result only one of the measurements could be used. FOUNDATION Fieldbus and PROFIBUS PA are fast enough to communicate many measurements per device in real time. For example, a density transmitter may provide the measured density, process temperature, and the computed referred density (figure 2-4). A single temperature transmitter with two independent inputs can be used instead of two transmitters.

Valve positioners transmit to the host a feedback of the actual valve position and the actuator pressure over the same two-wire network on which the valve setpoint is received but without the need for additional cables. Similarly, for many applications valve position alarms can be configured in software and communicated

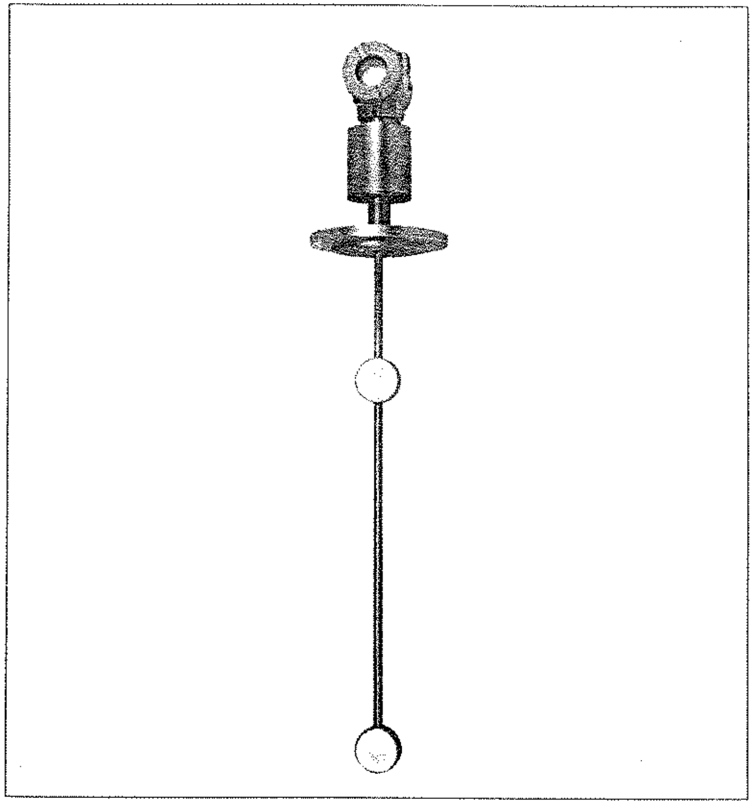


Figure 2-4. A multivariable density transmitter provides three measurements. (Courtesy of Smar)

over the network, without the need for external proximity switches or additional wiring.

Network-enabled Asset Management

Fieldbus technology is more than just the transmission of process I/O. An important advance is that other non-I/O data in the system rides the bus. This data includes configuration download, remote operation and tuning, and asset management functions like diagnostics and calibration data. Traditionally, asset management has been stand-alone software that used manually entered data. Fieldbus communications enables the asset management software to talk directly with devices throughout the plant. All the information about the device is visible from the host, including tag and other identification, ranges, materials of construction, last calibration record, and so on.

Network-enabled asset management features are therefore extremely helpful in both commissioning and maintenance tasks. In a system based on fieldbus technology the operator can look into every fieldbus device in the plant from the workstation. This was never possible in the past because conventional systems could only handle analog device I/O without providing access to intelligence in the field instruments (figure 2-5). Usually, configuration would be done using a handheld terminal connected to the device wires. A control system that only accepts process variables from its transmitters and sends out nothing but manipulated variables to the valve positioners provides only a fraction of the benefits field instruments could provide.

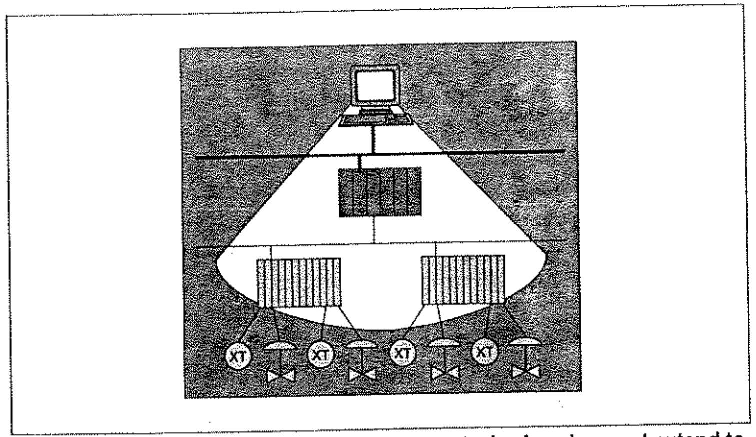


Figure 2-5. In conventional systems the operator's view does not extend to the field devices.

Digital communications means that all the device information can be accessed remotely over the network. It is possible to check on the calibration status of a device, set new ranges, and reconfigure and diagnose the equipment without leaving the computer workstation. The greater the distance involved the greater the advantage digital communications offers. Within a small site like a plant this is a great feature, but for a long pipeline or scattered oilfield the benefit is even greater.

Decentralized Functions

Devices based on fieldbus technologies perform much more than simple input and output functions. A transmitter not only does a pure measurement but also the associated compensation and totalization. A valve positioner not only positions the valve but performs flow characterization and process loop control, among other functions. HART, FOUNDATION Fieldbus, and PROFIBUS PA all support additional functions like flow totalization in the field device, which can be accessed remotely for configuration, reading, and the like.

Devices that are based on FOUNDATION Fieldbus because of the programming language may have function blocks linked together for a wide variety of functions such as control, selection, alarms, limits, and computation. This makes it possible to form powerful control strategies that can be executed in the devices at no additional cost. For example, FOUNDATION Fieldbus positioners perform control in the field by executing the PID algorithm function block and offloading centralized controllers, thereby improving control. Devices that have dynamically instantiable function blocks as opposed to fixed-set function blocks make even more powerful schemes possible by allowing many blocks to be selected from a larger library and also several blocks of the same type.

Wider Diagnostic Coverage

HART, FOUNDATION Fieldbus, and PROFIBUS PA networking all bring out more data from the field devices to a greater or lesser extent and all have special parameters for diagnostics. Detailed diagnostics is perhaps the most important information that can be accessed in a device. Self-diagnostic routines check the health of the device and tell the rest of the system if the information coming out of the device can be trusted or not. Detecting device failure through advanced self-checks is the first step in a long chain of fieldbus functionality that results in savings throughout the plant lifecycle.

The fieldbus technologies have standardized methods for disseminating diagnostic data throughout the system and ensuring that it reaches the operator and in some cases can be used for shutdown interlocks. Diagnostic parameters have been standardized, giving a whole new level of visibility without custom programming. The operator can from a single workstation access all fieldbus instruments on the site at any time using the engineering and maintenance tool. The operator may also see faults automatically without the need for manual interrogation in the field using a handheld terminal. For example, if the supply air to the positioner is lost it is indicated on the workstation. The health of any fieldbus device in the system can easily be checked without having to venture into the field. As a result, a device that is not trusted can easily be investigated.

Analog systems do not access field instrument diagnostics. Because of the lack of standard communications protocols control systems have previously been unable to benefit from this depth of diagnostics. Interrogation was done from a simple handheld terminal in the field and even then was only possible with devices of the same brand. Because there was no standard way to access diagnostics until fieldbus technology was introduced, devices until recently have had only rudimentary diagnostics. Fieldbus technology has changed this, and manufacturers are now competing for the widest diagnostic coverage. Today, diagnostics is detected with a finer granularity and detail than in conventional devices and indicate failure and the severity of the fault. The internal diagnostics routines in the devices are, of course, proprietary and specific to the sensing technology used. However, the fieldbus protocols specify how the diagnostics shall be presented and interpreted by DD. As a result, devices and hosts can be mixed and matched and the diagnostics still seen.

In both PROFIBUS PA and FOUNDATION Fieldbus diagnostics is dynamically updated as the quality in the status part of every measurement and manipulated parameter. The status tells the operators and engineers the validity of the process value, making it possible to make decisions with greater confidence. In some systems, the status is used by the control strategy to take special action in the event of abnormal conditions.

Operational Statistics

Even better than being instantly notified of problems is being forewarned about problems to come. Such anticipation can be had much more easily and accurately by using the leading indicators of the factors that influence wear and tear. Most fieldbus devices measure the ambient or sensor temperature, which may be used to determine if the device is operating within its limits. Valve positioners collect statistics like cycle count by tracking the number of valve stem reversals, and they have a built-in odometer to anticipate performance degradation caused by increasing packing friction.

Simplicity

Many of the features of fieldbus devices enhance their sheer convenience and ease of use in comparison to traditional analog equipment. Many functions are automatic or can be performed remotely from a workstation without having to send someone into the field.

Many mundane tasks are eliminated or reduced. These simplifications often translate into savings too.

Common Look and Feel

A great advantage that follows from interoperability is that all devices, even though they are from different manufacturers, can be configured from the same configuration tool. This means that only a single software application has to be purchased and learned. Because every device has a logical grouping of data into blocks and the electronic device description, the data for all devices are presented in a consistent manner, which can even make working with new devices intuitive. This feature compares favorably to the past when each manufacturer had its own proprietary protocol. This forced the sites to retain half-a-dozen handheld terminals, interfaces, and software so they could configure the site's different devices, each of which operated in a different way.

Plug-'n'-play

The communications protocol part of the FOUNDATION Fieldbus includes a system management function. A major benefit it offers is that it automatically detects, identifies, and assigns the address to devices as they are added to the network. This reduces labor significantly, eliminates human addressing errors, and instantly indicates if the device is properly connected to the correct and functioning network.

Homogeneous Environment

Because FOUNDATION Fieldbus is both the communications protocol used for device configuration and the programming language used to build the control strategy, it is possible for a host to have a single integrated tool that can perform both functions. A single environment for configuring and troubleshooting devices and strategy simplifies work. With an integrated tool it is easier to keep devices and strategy in sync, without much manual intervention required and without leaving the console. This compares favorably with traditional proprietary language solutions in which configuring devices and building control strategy are done in different applications using a variety of configuration languages and methods, often in different workstations or more typically on a handheld (figure 2-6). Inconsistencies between the controller and device databases can easily occur.

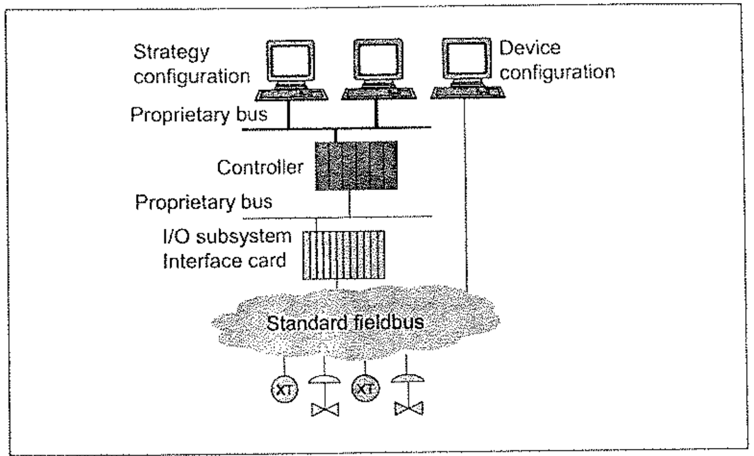


Figure 2-6. A system with proprietary language most likely has stand-alone device configuration.

Powerful Function Blocks

The standard function blocks that make up the FOUNDATION Fieldbus programming language are extremely powerful. Each one typically groups the functionality corresponding to several blocks in proprietary languages into a single block. However, what really makes these blocks powerful is their ability to handshake and propagate status along with the values that are passed from one block to another. Because block behavior is standardized these functions work across the devices of different manufacturers. The blocks contain standard shutdown interlocks and cascade initialization mechanisms using the status. This means that additional logic in different languages need not be configured to achieve these and many other functions. In other words, sites can take advantage of the power not just of the individual block but the combined power enabled by the standardized interoperation of links between these blocks.

Accuracy

In a system based on 4-20 mA the signals in a control loop undergo several analog-to-digital (A/D) and digital-to-analog (D/A) conversions. A transmitter using an analog output will have to do a D/A conversion to get the 4-20 mA output transmitted to the DCS I/O subsystem. The analog input module in the DCS will do an A/D conversion, which will allow the DCS controller to process the signal. An analog output module does a D/A conversion to get a 4-20 mA output signal. Lastly, a smart positioner must have an

A/D converter to allow its microprocessor to accept the analog signal.

Each conversion adds an error called the "quantization error." The errors from the four stages of conversions add up, often exceeding 1 percent. Digital networking eliminates the need for these conversations as all microprocessors communicate with each other. This results directly in increased accuracy. Keeping the signal digital from the very beginning to the very end makes possible infinitely more complex and precise signal processing. In HART, FOUNDATION Fieldbus, and PROFIBUS PA all scalar values are processed, stored, and communicated using the standard 32-bit floating-point format while maintaining seven-digit precision throughout the system. Since the values are floating point and typically in engineering units there is no need in the vast majority of cases to enter scaling or ranges. This simplifies as well as reduced the configuration effort.

Achieving Fieldbus Savings

The fieldbus technologies have features that lead to huge initial and long-term savings, but it is easy to miss the big picture. Although the savings on cable are the most visible, they are neither the only nor the greatest form of cost reduction that can be achieved using PROFIBUS, FOUNDATION Fieldbus, and—for many points—HART (provided the host has permanent communication). The initial savings include lower costs for purchase, engineering, and installation. The long-term savings include lower costs for maintenance and operation, thanks to network-enabled asset management but also lower costs for expansion and change.

Cost savings can be substantial, but, of course, they vary with the labor costs of the country in which the plant operates, the plant's size, and the nature of the plant. There is no standard way to calculate savings. The savings in cost and time for the site may be analyzed in terms of its unique conditions based on the possibilities offered by the fieldbus technologies. Because long-term savings are greater than procurement and installation savings fieldbus equipment pays for itself very quickly. It therefore also makes sense to re-instrument an existing site with fieldbus. PROFIBUS and FOUNDATION Fieldbus devices are more expensive than HART devices, which in turn are more expensive than their conventional counterparts. This fact offsets some of the savings fieldbus provides, but overall there is still a significant cost saving.

Lower Cost of Purchase

Because of its leaner architecture a system that is based on fieldbus technology in general but has FCS (Field Control System) architecture in particular requires a site to purchase and install significantly less hardware than a traditional system. Fieldbus-based systems are therefore lower-cost systems. However, the manufacturers' development cost for conventional systems has usually long since been amortized, and when controllers and margins are minimized the prices can come down low. Nevertheless, a conventional system will be more expensive to operate in the long run, resulting in a higher total cost of ownership.

The savings apply to the part of a system that is purpose-built for fieldbus technology. Other sections of a site that use conventional technology will reduce the average savings. Some systems may be built on architecture that prevents some savings from being achieved. Each installation is different, and therefore the total savings vary, but significant savings can be expected.

Fewer Parts

With fieldbus technology, reduction in hardware is seen at all levels of the system architecture, from field instruments to the workstations. Fieldbus technology largely eliminates the need for the I/O subsystem. As the number of components that are needed is reduced the panels they are installed in also become fewer and smaller. Thus, the large panels associated with traditional systems no longer need to be so large. Fewer, smaller, and lighter components also mean reduced shipping costs.

Because fieldbus-based systems can fully utilize multivariable transmitters and valve positioners the number of devices and associated wiring and I/O can be reduced. For example, in a density measurement on a tank a single transmitter can replace as many as three transmitters. A single temperature transmitter with two independent inputs reduces the number of transmitters to half, resulting in a 50 percent cost saving. The only means analog systems had to detect device faults was using discrepancy checking to compare the signal from two redundant transmitters—a solution too costly for all but a few critical control loops. Devices using fieldbus technology detect their own fault and communicate their health. As a result, one device can be used instead of two, reducing costs by 50 percent for such loops.

Since many devices can be multidropped on a single pair of wires the amount of cable can be drastically reduced. The amount of cable reduction depends on the wiring scheme, the number of devices installed on each network, the distances, and whether or not the installation is in a hazardous area. The main cable reduction is seen on the multi-core homerun cable from the field junction box to the I/O marshalling panel (figure 2-7). By putting twelve devices, some for control and some for monitoring, on each network the cabling is already reduced by more than 90 percent, which still leaves room for more devices. The cost reduction is significant, especially for armored cable. Additional related hardware savings are gained since fewer conduits, cable trays and ladders, and cable glands are required than in a conventional solution.

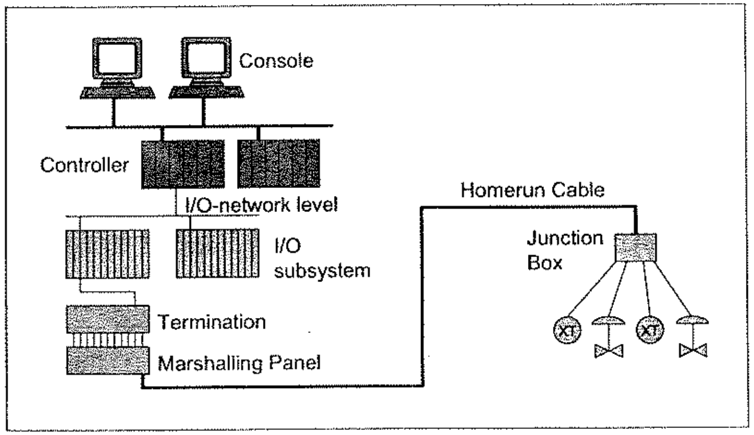


Figure 2-7. Traditional system architecture requires lots of hardware.

Using the multidrop capability it is possible to connect several devices to each safety barrier, thus reducing the number of barriers required in an intrinsic safety installation. The number of devices per barrier depends on the devices' power consumption, the barriers' current capacity, and the intrinsic safety method being followed, among other factors. Using the traditional entity concept that consider the cable a concentrated capacitance and uses a linear barrier some four devices can typically be connected to each barrier, but using the Fieldbus Intrinsically Safe Concept (FISCO) this often increases to as many as eight. At the same time it also allows for longer cable. That is, in terms of quantity the reduction in cable is 75 to 87 percent, but because of the price difference the dollar savings are slightly lower compared to a conventional barrier. Cable length is a critical factor since longer cable allows the safety barriers to be mounted outside the classified area where special enclosures are not required. This is explained in detail in chapter 3, "Installation and Commissioning."

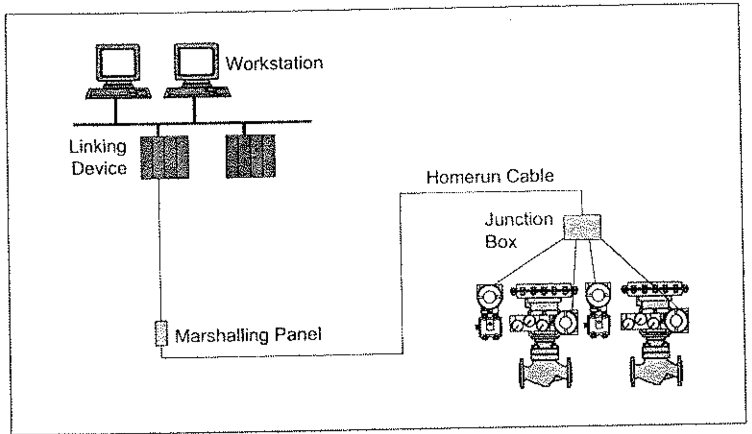


Figure 2-8. Fieldbus architecture reduces hardware.

Similarly, the marshalling panel also becomes smaller and simpler because the number of wires is reduced as a result of multidropping (figure 2-8). When digital communication is used no costly analog input and output modules are required. A typical scenario would consist of a single linking device processor module with four integral fieldbus ports connected directly to 64 devices. Such a scenario reduces the cost significantly compared to a conventional system that would require eight analog I/O modules of eight channels each. The simplification is greater still for the loops for which redundancy is required. Depending on the fieldbus-versus-conventional mix the analog I/O reduction can approach 100 percent. As the number of modules is reduced, the associated racks, nests, I/O network, and other mounting hardware are eliminated as are the power supply components for all the modules. The termination cards used for conventional I/O are not required for fieldbus communications. The I/O subsystem for fieldbus devices is therefore very much simpler and less expensive than for a conventional solution.

In a system based on a standard fieldbus technology the need for gateways between different network protocols is greatly reduced and often eliminated. Similarly, there is no need to develop custom device drivers, which removes a large expense.

Using the FOUNDATION Fieldbus capability to distribute control strategy execution into the field devices enables sites to drastically reduce the amount of centralized CPU processing power required. Hence, the number of controller modules can be cut since they are basically only needed for the less computation-intensive discrete control tasks. Since these are expensive parts this elimination results in great savings.

Less Space

The reduction in hardware means smaller panels and consequently a smaller space requirement. Thus, the need for climate-controlled rack rooms for the panels is reduced too. Since the wiring is reduced there is often no need for elevated floors. This can add noticeably to the savings, especially on rigs or other vessels where space is at a premium. In some cases, it may also mean that a new rack room need not be constructed for an expansion.

Less Expensive Parts

Sites that invest in a system based on open standard fieldbus technology further their lower system cost by not being tied to a single manufacturer. Independent of which host is chosen for the system, the field instruments and other system components and software can be sourced from a variety of other suppliers. Selecting the device with the best cost-performance ratio is recommended. Open competition lowers prices. This compares favorably to traditional proprietary systems where field devices had to be bought from the host vendor in order to achieve any integration. Having locked the customer in the supplier was in a position to charge higher-than-normal market prices.

Host-level networks based on Ethernet use standard interfaces and other network hardware that utilize standard communications software embedded in the workstation operating system. Ethernet components are so-called COTS (commercial off-the-shelf) products, which are available just about anywhere at extremely competitive prices, pushed down through the economies of scale that can only be attained by sales volume in the millions of consumer goods units per year. Compared to proprietary networking parts, which are assembled only in the hundreds, Ethernet is much cheaper.

Less Spare Capacity

Adding control loops or monitoring to a conventional system meant adding lots of hardware. Installing additional equipment to accommodate later modifications from the original plan was troublesome and disruptive, hence the system suppliers charged high rates for engineering changes. As a result, in the past systems were always purchased with lots of spare capacity for I/O modules, rack slots, controller memory, and processing capacity. This over capacity meant that plants were buying a system far bigger and costlier than they initially needed just to be on the safe side.

Expanding or changing fieldbus-based systems is much cheaper and simpler because they have much less hardware, hence large amounts of spare capacity need not be purchased at the time of initial investment.

Fewer Spare Parts

A lean system has less hardware that can fail in the first place. Therefore, fewer spare parts need to be kept in the inventory, which further reduces the overall system cost. Standards-based equipment is provided by many second sources, so if a spare transmitter cannot be delivered in time from one supplier another brand with shorter lead time can be used in its place. As a result, keeping in-house spare stock on hand is less critical, and expensive contracts for the manufacturer to bond stock may be avoided.

Engineering Savings

The simpler hardware architecture and reduced complexity of a fieldbus system greatly simplifies project engineering and design work from the conceptual design stage, through the detail engineering and documentation phases to training and so on. Fewer engineering hours spent means lower cost. Projects can be run "fast track" to get the plant up and running sooner and thus to start paying back the loans.

Faster System Design

For fieldbus devices it is easier to define host I/O requirements than for conventional systems. All that is required to get an estimate of communication port count is an estimate of how many fieldbus devices the system shall support. Unlike conventional systems all fieldbus devices are alike, so it is not necessary to specify if FOUNDATION Fieldbus or PROFIBUS PA devices are analog or discrete, input or output, single or multiple variable, or used for monitoring or control. In these cases, there is no need to select different types of I/O modules either, thereby reducing conceptual design time. The detail design phase is also faster as, for example, calculations for I/O module power consumption and the like need not be done. Simpler panels are designed faster.

In the case of FOUNDATION Fieldbus the functional requirements for control are also much easier to define. Previously, it was necessary to explicitly define every required function in lengthy specifications, for example, that the PID block should have functions such as reset windup protection, cascade setpoint, override, deviation alarm, and so on. Now, the programming language defines the standard function block features, so the only thing that need be specified is which one of the blocks is required since the block features are implicit.

Faster Vendor and Device Selection

In the past, selecting a particular brand of host and controller for a system meant that the bulk of the field instruments had to be bought from the same supplier to ensure that they would work together. Selecting that one supplier was a difficult compromise—for example, choosing between the most powerful controller, the most accurate transmitter, or the quietest valve. Controller features and device performance for different systems had to be weighed against each other. A standard fieldbus simplifies the selection process since the best devices of each type can interoperate with one another, which makes it possible to finish the bill of materials faster. Moreover, one type of cable is used for all fieldbus devices, which reduces the types that must be chosen. Similarly, a single safety barrier type is used for all fieldbus devices, eliminating the tedious process of comparing entity parameters. When multivariable devices are used instead of individual units the number of devices that need to be specified is reduced.

Faster Device and Strategy Configuration

Thanks to fieldbus technology, configuration development and entry time at all levels of the system from field device to host is reduced. For example, ranging is not necessary for field devices in most applications because HART, PROFIBUS PA, and FOUNDATION Fieldbus use engineering units when they digitally communicate a variable rather than scaled ranges, such as 4-20 mA or 0 to 100 percent. Unlike a conventional system, if the value in fieldbus is 200 kPa, two hundred is transmitted, not *x*% or *y* mA. Since scaling is eliminated device configuration and verification work is reduced. All device settings such as sensor type selection can be made off line before the device is manufactured and downloaded during commissioning. This allows configuration to start immediately upon project kickoff.

The analog I/O modules are essentially eliminated, and therefore there is less need for I/O-subsystem configuration. This saves time in comparison to a conventional system where racks, modules, and channels must be configured. I/O module auto-detection is no solution for projects on the "fast track" since the engineering will often be completed before the hardware is manufactured. Thus, eliminating I/O altogether is the fastest solution. A standard fieldbus protocol eliminates the need to program proprietary protocol drivers and map custom data, which are time consuming, subject to error, and cause costly delays in projects.

For a host using FOUNDATION Fieldbus the standard programming language facilitates the development of control strategy and reduces configuration time. Many of the interlocks require absolutely no additional configuration effort (see chapter 4), which minimizes configuration entry time. Thanks to a standard language the system is truly open, which allows anybody to learn how to configure the control strategy without the need for expensive services from the supplier, as is often the case for proprietary solutions. Furthermore, some hosts have the device and control strategy configuration integrated into one and the same tool, which eliminates the need to switch between applications, enter the same information twice, or even work on two different workstations. The automatic device identification and address assignment makes possible configuration work that is based entirely on tag that is more intuitive, less subject to error, and infinitely faster than the mapping of device addresses and memory registers, as in older systems.

Less Documentation

Reduced hardware means that project documentation is also reduced. Using multiple variable devices instead of many single-point units means that fewer instrument specifications, indexes, and data sheets have to be generated. Multidrop wiring means that one single network drawing can be made per network instead of sixteen individual loop drawings as in a point-to-point scheme. This reduces effort by as much as 90 percent. Since the I/O subsystem is gone there is no associated cross-reference listing. A copy of the fieldbus device configuration data, such as sensor types and ranges, is stored in the configuration tool instrument database ready for printout or export to, for example, Microsoft® Excel. During the commissioning, the changes made are also stored, creating an up-to-date "as-built" configuration database electronically. Such documentation features were not possible in DCS where a handheld was used for device configuration. Smaller control and marshalling panels that have less equipment in them have simpler layout drawings and wiring diagrams.

Construction Savings

The less hardware there is to install and wire up, the greater the savings.

Faster Fabrication

Reductions in the amount of equipment means smaller panels that have less marshalling, termination, I/O modules, and power suppliers inside. This makes them faster to build and wire. Using a linking device with built-in device power and processor results in a solution that is significantly more compact and faster than conventional I/O.

Easy System Verification

Before fieldbus, the I/O-subsystems calibration check of every I/O point was a long and tedious part of the verification process that was required to ensure that termination, fusing, I/O modules, and racks were functioning properly. Since these parts are eliminated in a fieldbus system there is no such test, and the total verification time is drastically reduced. A simple check of the communication to verify the integrity of the communication port is sufficient. Since values are transmitted in engineering units there is a much less pressing need to check the consistency of scaling between devices, controllers, and workstations. This is also a key time-saver since in the past such checks were made from different software tools and a handheld. This characteristic of fieldbus technologies reduces the risk of alarm and shutdown trips not working when they should. Since communications is based on tags, the process of verifying parameter mapping to the device address and memory registers employed in other protocols is done away with.

Less Installation Works

Multivariable devices cut the number of devices to be installed for certain application by a factor or two or three. Fewer pipe stands and process penetrations mean a reduction in installation time. Depending on the ambient conditions the associated heat tracing or shades are reduced too. With fieldbus it is possible to install devices where they previously would never be installed because gaining access to check would have been too difficult. Because fieldbus allows remote diagnostics the devices will be physically checked less frequently, which makes easy access less critical. Piping can therefore be simplified by, for example, installing directly at the top of a vessel without wet leg and pots. Improved performance may be an additional bonus. Since the reading from a hard-to-reach or -see device can be transmitted to any other device, it is possible to have the indication on a more visible device, if field display is desired.

Less Cable Run and Termination

The fantastic reduction in wiring made possible by fieldbus device multidropping and multivariable capability means that fewer cables need to be pulled. Using fieldbus reduces what for conventional systems was a major logistical effort into a simple task. Consequently, fewer conduits are required for protection or to make the installation flameproof. A minimal number of cable trays and ladders are required to support these cables. The man-hours spent to install all of these elements is far smaller than what would be required for a conventional wiring scheme. Reduced wiring also means fewer terminations, that is, fewer wires to cut, strip, crimp, label, and connect. By using fieldbus, as many as ten terminations per point can be eliminated, counting from device to marshalling panel. Given that each termination takes a couple of minutes this becomes a major cost reduction in comparison with a conventional system, especially in view of the large number of devices found even in a medium-sized installation.

Faster Commissioning

Unlike a traditional system, where commissioning is a very manual and labor-intensive task, a fieldbus device, once connected to the network, can be commissioned from the workstation in the control room. This makes the commissioning procedure very simple and fast, and for the most part it can be carried out by one person. Once the host is connected to the fieldbus networks, all field devices on the network are automatically detected in one go and the tags identified in the host's "live list" for each network. This is far easier than the "ringing out" of instruments one by one that is performed for conventional systems. That process requires people in the field equipped with simulators who maintain radio contact with the control room to verify correct wiring.

The asset management features not available in conventional systems are very helpful during the commissioning process. It is easy to identify a device by checking the manufacturer, device type, versions, tag, serial number, and application descriptor. Likewise, the fieldbus device configurations are verified from the workstation,

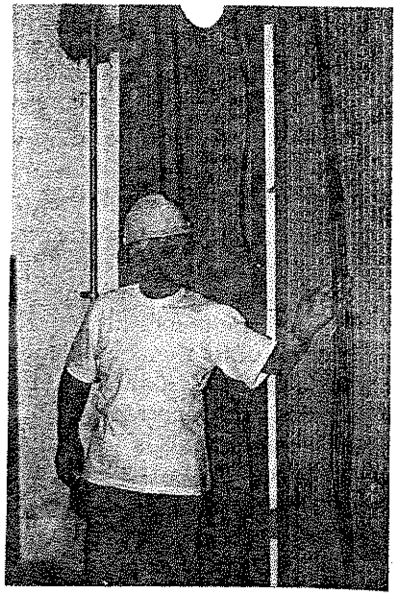


Figure 2-9. Fieldbus reduces wiring. (Courtesy of Smar)

and any configuration or range changes that have been done are automatically recorded in the database "as built," which maintains the documentation as ready for printout or export. The configuration for all devices can be downloaded with a single click. In the past, changes made using a handheld terminal needed to be manually updated. The chances of spotting a range mismatch between the transmitter and the controller were slim, which opened the possibility of a wrong reading and subsequently a nonfunctioning control loop and trip points.

Remote commissioning of sixteen devices at a time speeds up the process. The simulation function may be used to verify that the system has the correct response by safely testing normal, abnormal, and dangerous process conditions. This can easily be done without needing to apply any physical input and even without using a simulator, which was a requirement in conventional systems.

Maintenance Savings

The cost of maintenance can be reduced in many ways, e.g. by preempting failure but without unnecessary expenditure on preventive maintenance and by being better prepared when something does occur. The advanced self-diagnostic tests and operational statistics information that thanks to fieldbus technology can be retrieved from devices throughout the plant may be used for asset management to anticipate failure, suspect drift, confirm malfunction, and verify configuration etc. These features are embedded in devices and software requiring little or no configuration making it possible to benefit directly without the need for expensive consultants or long implementation periods. Maintenance savings can be achieved provided that the maintenance tool is an integral and permanent part of the system so that diagnostics is continuous and asset management functions can be carried out simultaneously with normal operation. Handheld terminal or temporarily connected software is an obstacle to efficient practices. This form of network enabled asset management was not possible before fieldbus and is not available in traditional systems.

Less to Maintain

A leaner and less complex system based on fieldbus technology has fewer parts that can fail in the first place thus there is less parts to maintain. The elimination of intermediate 4-20 mA signal means that there is no need to check and calibrate I/O cards, transmitter current outputs and positioner inputs.

Reduce Field Trips

In a networked system several maintenance task previously requiring a local field trip with the handheld terminal can instead be carried out remotely from the workstation. The computer therefore becomes the most powerful maintenance tool reducing time-consuming field trips and cutting maintenance cost. The host retrieves information from the field devices for asset management and dissemination throughout the enterprise independent of device manufacturer. Traditional systems with no or proprietary field instrument communication do not have this capability.

In a fieldbus host simple status is typically shown in operator screens appearing as soon as the device diagnostics detects the fault. Detail diagnostics is done using a maintenance tool. From the "live list" it is easy to tell when a device is disconnected, lose power have communication problems or fail. Remote performance testing exposes weaknesses and future problems e.g. it is easy to test if a valve is able to respond to small demands by actuating the valve in small steps, perhaps a fraction of a percent, while at the same time remotely monitor that the valve is really moving. This was not possible in the past, as most positioners did not have actual position feedback to the control room. Many problems can be solved remotely by simple configuration changes such as direct or reverse action, damping, or tuning. Remote maintenance is not a luxury it is a necessity for savings.

Preempt Failures

Network-enabled asset management features allow a proactive maintenance scheme in which operational statistics is used to estimate wear and tear. Together with information about exceeded operating conditions this wear-and-tear data determines how much longer a device can operate before requiring maintenance. It is possible to more accurately anticipate and schedule a failure or degrading device performance before it causes inaccuracies and faults but not unnecessarily early. Determining if a device may need to be repaired soon or even now based on the manufacturer's life expectancy data for critical parts minimizes costly surprise shutdowns and reduces unnecessary maintenance cost. Thus, fieldbus drastically reduces surprise shutdowns caused by unforeseen failures and makes it possible to plan maintenance for many devices on one convenient occasion. This could not be done by systems in the past.

Faster Repairs

Because the fieldbus network communicates defined faults more precisely immediately upon detection technicians can more accurately pinpoint the source of the problem, pick the right replacement parts and tools before going into the field, and thus solve the problem faster. Fieldbus also makes it is easy to retrieve information on a device's materials of construction when selecting a suitable replacement part for the application. Traditional systems do not provide sufficient level of diagnostic detail. Moreover, when technicians have installed replacement devices they can download their configuration and put them into operation very quickly.

Timely Calibration

Drift occurs over time and may also be caused when ambient condition limits are exceeded. A record of the last calibration is stored in each device where it will not be misplaced. Operators may review the record to determine the calibration status of the device to ensure that calibration is neither overdue nor done unnecessarily early. Sensor temperature can be monitored, and if drift is suspected because limits have been exceeded calibration can be performed earlier than originally scheduled. Positioner self-calibration for valve travel can also be invoked remotely. In fieldbus, it is easy to gain an overview of which devices in the plant are due for calibration. Conventional systems do not enable such calibration management.

No Unnecessary Maintenance

Detailed device diagnostics makes it possible for operators to quickly determine if a process problem is caused by the device or the process. In the past, technicians were often sent into the field on false alerts to bring back an instrument for testing and then only find that it was OK. Using fieldbus, it is possible to remotely check if the problem is genuine before going into the field, thus maintenance is only performed when it is really needed. The time and cost saved by not having to bring only a few transmitters into the workshop is enormous. Less money is wasted checking and repairing devices that are OK. By referring, for example, to the number of reversals, a technician can decide whether or not to tear down a valve so as to replace a valve stem packing. Conventional systems provide no means for achieving a comparable result.

Market-based Replacement Part and Maintenance Prices

A system based on open-standards fieldbus also offers the benefit that failed components can often be replaced by components from other suppliers if the original parts are overpriced. Open standards technology means less reliance on a single supplier. It is not necessary to get tied up with the costly maintenance contracts associated with proprietary systems.

Operation Savings

Asset management does not deal directly with process control. Its focus is on improving maintenance, which indirectly improves control because when devices perform better and with fewer failures better plant performance results.

Less Process Downtime

It is difficult to operate any plant efficiently when device problems are recurring. Shutdowns caused by failures are normally very expensive because entire batches of products are often destroyed, and production capacity may be reduced for extended periods of time. When the process is not running money is being lost.

Maintenance schemes that use network-enabled asset management features result in fewer interruptions caused by surprise device failure shutdowns. When operators know about an imminent failure they can order replacement parts in advance. They can respond to faults faster and reduce repair times, which results in less plant downtime. Early warning makes it possible to solve problems before they adversely affect plant output. With equipment in good condition the process is kept on line longer. Systems that lack proper network infrastructure in the field do not offer these benefits.

An advantage fieldbus technology offers over traditional DCS architecture is that it minimizes the number of I/O modules, controllers, and other components, which reduced the probability of failure. The architectural simplicity of a fieldbus system thus contributes to its availability. There are simply fewer parts to fail.

It is prudent to prevent control for the loop from continuing if the transmitter fails. An instrument failure indication means that the loop will be shut down. Conventional transmitters and DCS don't have the means to accurately exchange status information because of the limitations of 4-20 mA. Typically, a transmitter output current below 3.8 mA or above 20.5 mA is the only fast way to indicate a fault and to make the interlocks shut the loop down. For analog systems, valve positioner faults are even more difficult to detect. Because analog systems make no distinction between severe and minor faults even a small problem would stop the process. Furthermore, an under range of a mere 1 percent would also trigger a stop. Such unnecessary trips disrupt production, which increases downtime and reduces profits. Fieldbus devices, on the other hand, are better equipped to communicate the severity of the fault and detect that the problem is genuine. This superior data validation results in a reduced number of spurious shutdowns.

In traditional systems, the I/O for as many as one hundred loops could be communicated on one bus from the I/O subsystem to the controller. Supervision for as many as a thousand loops could also be communicated on just one bus from the controllers to the host. The failure of such a bus would trip all those loops, undoubtedly halting most of the plant. The field-level network for FOUNDATION Fieldbus and PROFIBUS PA is highly distributed in comparison to the network technologies used at the I/O and control levels of earlier systems. This distribution leads to better availability. At the very most, eight but typically four loops rely on a single network. This reduced the number of loops that would have to be shut down in the event of failure. Production standstills are rare. Costly one-to-one redundant I/O subsystems and networking would be needed to achieve the same availability in a conventional system, yet in most cases the backplane and intermediate I/O switching boards would remain weak points.

In the past, control systems had centralized controllers that handled dozens or even up to a hundred loops. Control distributed to the field devices is one of the keys to the high availability of the FCS architecture that can be built using FOUNDATION Fieldbus. The FCS architecture is highly distributed: a transmitter and a valve positioner make up the typical loop, and the PID algorithm normally executes in the positioner, achieving single-loop integrity. Because there is no centralized controller a single fault does not affect any large number of loops. Costly one-to-one controller redundancy in addition to I/O redundancy would be required to achieve the same availability in a conventional system.

Less Process Variability

Factors like long-term drift, imprecision due to wear and tear, and poor tuning cause deviation and instability in process variables, which ultimately result in variations in the final product. Because fieldbus-based systems are completely digital, noise and many other sources of errors are eliminated. Operational statistics such as counting the number of reversals of the valve actuator also provide another major benefit: the ability to spot poorly tuned positioners and control loops. A valve with a large number of reversals is a clear indication that the loop is oscillating, meaning that retuning will be required. With fieldbus, it is thus possible to access this information in the positioner to identify the culprits behind variability and target the resources to fix them.

Fieldbus also allows critical parameters to be monitored in real time without needing to go into the field. This makes it possible to fine-tune the loop for optimal production output. Calibration management also promotes the timely checking and calibration of instruments to maintain accuracy. A control valve's capacity to respond to small demands is important for tight control and can be verified remotely. These features play a vital role in achieving a more uniform product because they remove the sources of variability one by one. They are not available in conventional system architectures.

Higher Quality

Tighter product consistency is achieved through asset management based on the diagnostics and operational statistics received over the fieldbus network. Such consistency ensures product quality within specified limits. Using the status and diagnostics retrieved from the devices enables operators to quickly stop production in the event of failure before any out-of-specification product contaminates good-quality product. Less time, raw materials, energy, and other resources are wasted to produce lower-grade product or product that cannot be sold. By using the FOUNDATION Fieldbus programming language such interlocks are built in. The improved calibration scheme made possible by network-enabled asset management features makes it easier to remain in compliance with quality and environmental management procedures. There are no alternative means to achieve the same result.

Greater Safety

Accidents carry high costs. It is obviously desirable to minimize the risk of harm to people, environment, and equipment. To avoid accidents control must stop when a failure occurs.

WARNING—For those sites that have process units that have been assessed as having high risk or hazard an approved safety-related system of a suitable requirement class should be used.

For regular controls, fieldbus technologies have several characteristics that sites may use to improve the safety of basic control systems throughout the lifecycle to a level well above that of a conventional system.

Undetected faults are very dangerous because the system is not shut down. In an analog system, a fault often goes unnoticed, and dangerous control may continue because of an untrue value from a failed transmitter. With fieldbus technology, such a failure is detected and communicated, which allows the loop to be shut down. Diagnostics is indicated as part of the status that is communicated with the measurement and control variables passed between devices. Because there is a distinction between minor "uncertain" and severe "bad" problems unnecessary shutdowns can be avoided using fieldbus.

Interlocks may initiate a shutdown by using the status received with the process variable from the transmitter in the basic control strategy. This can be done by setting the status on the manipulated variable to the valve positioner accordingly. If the FOUNDATION Fieldbus programming language is used, such shutdown logic is already built in and only needs to be enabled. Thus, the field devices themselves automatically ensure graceful shutdown without the need for a central controller. For example, a sensor failure such as a burned-out thermocouple would be detected by the self-diagnostics of the temperature transmitter, and a bad status would be propagated through the control strategy. Control would stop and status would be further propagated to the valve positioner, which would move to a predetermined position. Similarly, the device diagnostics detect bad communication, which allows shutdown action to be taken. Therefore, multidropping devices on a network is not dangerous. Manually configuring and verifying such a safety in a conventional system would be complex and conducive to error. The crude diagnostics in traditional systems makes it difficult to balance safety and availability. Using fieldbus technology for basic controls is a good idea because it enables some of the features and characteristics previously only found in a full-fledged safety-related system to be used in basic control.

WARNING—As has been the norm before, valves should still be selected with mechanical fail-open or fail-closed as an additional level of safety. This provides a last line of defense in the event of power failure.

Only safety-approved devices based on safety-approved protocols should be used with safety-related systems. One such networking technology is the PROFIsafe profile of PROFIBUS, which can be used for communicating between safety-related field devices and a controller. The PROFIsafe technology has received favorable reports for use at safety level SIL3 (IEC61508) and AK6 (DIN V 19250) without using redundancy since redundancy is only required for availability. The mechanism for safety is implemented as software functions on top of the standard transmission system. Because no modifications are made to the standard protocol, safe and standard communications can be mixed, and both types of devices can therefore share the same wire. Safe transmission includes several measures for discovering faults on the standard transmission system that have been caused by noisy line or hardware or software failures. Errors that have to be detected include repetition loss, insertion, incorrect sequence, corruption, delay, mix-up of safety-related and standard messages, and erroneous addressing. Measures taken in addition to the normal communication checks include sequential numbering and check of messages, comparing delays to expected time, authentication of messages, and additional data integrity checksum. At the time of writing no PROFIsafe products have yet been made available. Other safety-related bus technologies exist, for example, technologies for simple discrete I/O and also proprietary protocols for safety-related controller communication over Ethernet.

Less Training

Training is costly, but standards make it possible for devices and software to function and be operated in the same basic way. This reduces the need for ongoing training of personnel in the use of different equipment and tools while at the same time reducing confusion and mistakes. Interoperability makes it possible to use only a single tool to configure devices from different manufacturers, giving everything a common and user-friendly look and feel. FOUNDATION Fieldbus also specifies a programming language, which makes it possible to build a control strategy using controllers from different manufacturers and then executing the strategy in a transmitter, positioner, or centralized controller.

Lower Cost of Expansion and Change

For the same reasons that fieldbus-based systems are cheaper to buy and deploy, they are also cheaper to expand and modify. However, there are additional benefits that make them even more advantageous in this respect.

Scalability

Increased product demand may force a plant to expand, and a production process improvement may require that some additions be made. If a network is not fully loaded field instruments can be added even without pulling new cable. Each leftover communication port can handle sixteen new devices. Expansion becomes both easy and cheap. Building large systems is a matter of linking more field-level networks to the host-level network.

The FCS architecture is the key to the ability of a system to expand at little additional cost because there are no expensive central units. In a traditional system, when a site adds loops an additional I/O subsystem or controller will soon be required because there aren't enough I/O channels or processing power. Adding that loop may be very expensive, which may cause an idea to be cancelled or put on hold. Simple improvements that were previously prohibitively expensive become feasible using bus technology. The fundamental limitation of the scalability of the traditional system architecture is that its resources are drained as loops are added. Every instrument added takes away some of the finite resources of the system. The limits for controller memory capacity, I/O counts, controller address range, and the like are approached. As CPUs get loaded with more and more loops to keep costs down, performance suffers. Another restriction to be aware of is that older proprietary protocols often have limited bandwidth of perhaps 500 kbit/s for the I/O network and 1 Mbit/s for the control-level network. Ethernet at the host level running 10/100 Mbit/s overcomes this limit.

In contrast to the traditional system, adding a device in an FCS means adding resources. In the FCS, adding instruments means more CPUs, more memory, and more computing power in the system. Though it is counterintuitive at first, the more the system is "loaded" the more powerful it gets. Every device added is one more CPU sharing the workload. This means that the system can be continuously expanded and handle ever more advanced control strategies without a corresponding loss of performance. Because the FCS architecture is built by smaller decentralized components sites can add instruments basically one by one without having to add major components. For the same reasons, systems can economically start very small and gradually expand as the demands grow. Similarly, since there is no heavy investment in a large central component, it is even economical to build a system with only a handful of loops.

Easy to Modify

A plant may have to adapt to market requirements by changing product and production methods. Such modification is easier in a fieldbus-system because new configurations can simply be downloaded, preparing the device for the new application. This provides a great deal of flexibility to respond to quickly changing production demands, material conditions, and equipment situations. Unused functions can be put into use at a later stage. For example, operators can put actual feedback from a valve position into use without having to modify the valve positioner, running wires, or adding any AI points to the control system as was necessary in the days of analog feedback. With a fieldbus-based system it is easier and more affordable to make small changes. Thus, small improvements can constantly be made without costly change orders to the supplier.

Protected Investment

A great obstacle to improvement may be finding that a system bought only a few years ago has been replaced by a new model that requires a complete replacement or expensive gateways, software upgrades, and services for integrating the new with the existing. However, once laid down standards don't change very quickly, and therefore manufacturers are forced to go that extra mile to follow the standard when inventing new features and improving performance so products remain compatible with existing equipment. The stability of the fieldbus technologies therefore protects the investment a firm has made in a control system from the obsolescence that often occurs with proprietary technologies. Future compatibility is ensured, and the threat of complete replacement and reinvestments is removed.

Fieldbus Doubts Addressed

Fieldbus technologies are somewhat disruptive because they change the way things are done and the tools required to do them. However, once implemented these changes are all for the better.

Discrete I/O and Control

FOUNDATION Fieldbus and PROFIBUS PA were specifically developed to meet the demands of the process industries, which primarily require measurements that are analog in nature and control that is modulating. However, because discrete sensors and actuators are an integral part of almost any control strategy their functionality was built in from the beginning. The field-level protocols are based on the 31.25 kbit/s option of the IEC 61158-2 standard. This is too slow for a "large" amount of "simple" discrete I/O, which it is not intended for. However, it is well suited for a dozen or so points per network. On/off valves and small discrete remote I/O modules for these technologies are already available in the market. An important point to keep in mind is that network technology that utilizes discrete I/O no longer has to be "simple." In addition to the pure input or output function, diagnostics can also be communicated from discrete devices, which therefore can be more advanced. This includes self-checks to verify misalignment or soiled optics for sensors as well as open/close verification and cycle counting for on/off valves. Primitive low-cost I/O network technologies do not have the capability for intelligent diagnostics, and therefore they do not provide long-term maintenance savings.

Another trend is the move away from discrete switches, for example, for pressure, and toward the use of transmitters even in simple trip applications. The reason is again that simple discrete devices provide no diagnostics, and it is therefore impossible to tell if the state really has not changed or the switch has failed. This uncertainty may lead to potentially dangerous situations. Transmitters

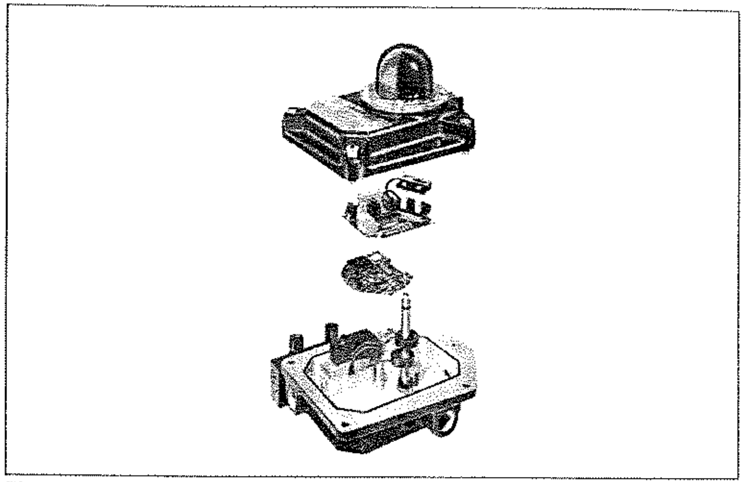


Figure 2-10. On-off valve actuator with FOUNDATION Fieldbus. (Courtesy of Flowserve)

are now relatively low cost and provide the diagnostics to validate the signal as well as other benefits, such as easily adjustable trip levels and tendency indication. Control valve positioners detect end of travel in software rather than with mechanical limit switches, which further reduces discrete I/O.

The solution for applications in which regular discrete I/O in high volume is still being used is conventional remote I/O on the host-level network, which has sufficient bandwidth for large volumes of high-speed data.

The function block programming language that is part of the FOUNDATION Fieldbus includes blocks for discrete control. Several analog blocks have discrete functionality as well. Because many of the required interlocks are already integrated into the function blocks, the amount of discrete I/O and logic is also reduced. A flexible function block that encapsulates logic that was created using, for example, any of the standard IEC 61131-3 languages makes it possible to configure discrete logic using the ladder diagram language, among others.

Migration and Integration

Most sites contemplating the introduction of fieldbus technology already have an existing system and components that have to be integrated with the expansion. Even in the case of a new greenfield project there is often one or more conventional subsystems that have to be linked up. A number of solutions already exist in the market, including converters for field instruments, gateways between networks, and OPC servers for systems. Old and new basic control, advanced control, safety-related control, and so on can therefore be integrated. Chapter 5, "Integrate and Migrate," covers these aspects.

What If the Cable Breaks?

A field-level network typically interconnects twelve or at most sixteen multidropped field instruments that provide power and communication on the same pair of wires. The cabling is not redundant, that is, there is only a single pair of wires, which of course means that cable damage results in a loss of communication and power for the devices on that network. However, for most practical purposes sites do not need to be concerned about the availability of multidropping, and several measures (presented in chapter 10, "Availability and Safety") can be taken to minimize impact. A cable is a passive component and therefore rarely fails on its own. The greatest risk to network integrity is during maintenance work when the wire may accidentally be disconnected or short-circuited. The scenario is similar to the situation in which an I/O module in a conventional system either fails as a result of over voltage, for example, or is mistakenly removed during maintenance.

A field-level network populated with sixteen devices supports at most eight control loops since a loop requires a minimum of one transmitter, one actuator, and often an additional device for an auxiliary function. Furthermore, some of the devices are used for monitoring and alarm. Therefore, there may be on average perhaps only five control loops per network. Because of the high level of decentralization only a few loops, which corresponds to a very small section of the plant, are affected by a single fault. The extent of the damage is no greater than that caused by an I/O module failure on a conventional system since one module there may have as many as 8 to 32 inputs (and much less than that in the case of an I/O backplane failure). In other words, because of the high level of distribution the failure of a field-level network is far less critical than the failure of a remote-I/O network or control-level network in a traditional system. By distributing I/O, a site does not need to use redundancy to achieve sufficient overall plant availability.

Plants normally identify a few critical loops around the site for which they need higher availability than other loops. The reason for this is that a failure of a critical loop would either require a shutdown for safety reasons or would result in great financial loss due to a production stop. For most plants, critical loops do not exceed 20 percent of all controls, and often less. In a traditional system, redundant I/O modules were required for the critical loops to ensure that they are still operational even if the primary module fails or is accidentally removed. The cost for one-to-one I/O module redundancy was more than double that of single I/O and in most cases made it prohibitive to use it for all loops. A common fieldbus-level network solution is to allocate only one critical loop per bus segment, with the rest of the devices on the network being used for noncritical loops or monitoring. Thus, if the cable fails, it affects only a single critical loop. This almost automatically becomes the case for every loop in an intrinsically safe installation where the network is divided up into several segments by repeating isolated safety barriers. This is particularly so if entity concept barriers are used since the power limits the number of devices to an amount roughly equivalent to one loop. That is, almost every loop has its own wire. However, the field-level network does support redundant linking devices and field power supply. In conventional systems, the wire pairs for sixteen or more devices were generally concentrated as part of the same single multi-core cable, and in case of damage to the cable many loops would be lost even if redundant I/O modules were used.

The more distributed the system the smaller the impact a failure would have on one of its components. Therefore, only a few field instruments rely on one wire.

As far as the availability is concerned multidrop networking is equivalent to multichannel I/O modules. However, fieldbus-based systems are safer because they provide better failure indication. They therefore compare favorably with conventional systems, which do not detect transmitter and valve faults.

What If the Controlling Device Fails?

What happens in a FOUNDATION Fieldbus-based system when control is performed by, for example, a valve positioner and that positioner fails? This situation is no different from that of a conventional control system today. If there is no positioner the process cannot even be controlled manually; the valve will go to its mechanical fail-safe position. If the positioner does not operate, having a controller makes no difference. In fact, a networked solution again fairs positively in comparison with a conventional analog solution because the latter's lack of diagnostics and feedback means it will miss the fact that the positioner has failed. Therefore, "control" will continue, unaware that the valve is not responding to the controller output. In a conventional system, the loop may thus appear to be OK when, in fact, the situation may be hazardous. Chapter 10 ("Availability and Safety") is dedicated to methods and practices for achieving high availability and safety.

The more distributed the system, the smaller the impact the failure of one of its components will have. Therefore, only one loop relies on each field device.

Ethernet Determinism

Earlier forms of shared-media Ethernet, such as 10Base-2 coax or 10Base-T with shared hub, are considered "nondeterministic." This is so because all the devices are fighting over the use of the bus, and sometimes two nodes access at the same time, causing collision and a random pullback time. However, by using switching hubs the network is not shared since the switch supports multiple simultaneous communications between different pairs of nodes. There is only one device per network, which essentially eliminated collisions.

The consensus appears to be that a network is deterministic if the chance of message delay is less than the chance of message loss as a result of noise, that is, one in ten million. Ethernet behaves deterministically up to some 50 percent loading for switched networks. This is quite easy to achieve for process control because the messages are short, especially when using 100 Mbit/s.

Fieldbus Comparison Table

The main features of the three field-level network protocols are summarized in table 2-1. The benefits that can be achieved from fieldbus depend largely on the technology chosen and how well it is implemented in the products chosen.

Table 2-1. Comparison of field-level network characteristics.

	HART	FOUNDATION H1	PROFIBUS PA
Version	5	1.4	3
Digital	Y	Y	Y
4-20 mA	Y	N	N
Intrinsically safe	Y	Y	Y
2-wire	Y	Y	Y
Multi drop	Y	Y	Y
Speed	1.2 kbit/s	31.25 kbit/s	31.25 kbit/s
Poll-response	Y	Y	Y
Publisher-subscriber	N	Y	N
Report distribution	N	Y	N
Device Description	Y	Y	N
Function block links	N	Y	N
Standard parameters	Y	Y	Y
Time syncronization	N	Y	N
Alarms	N	Y	N
Trend	N	Y	N
Scheduling	N	Y	N

Table 2-1. Comparison of field-level network characteristics.

EXERCISES

2.1 Do the Ethernet standards define the meaning of the data on the bus?

2.2 Who provides the device support files, the device manufacturer or the host manufacturer?

2.3 Are parameters associated with a particular type of sensing stored in a transducer block or function block?

2.4 Does diagnostics make devices safer?

2.5 Are fieldbus technologies that employ floating-point format or scaled integers used for analog values?

2.6 Does fieldbus require measurement ranges to be set?

2.7 Is FOUNDATION Fieldbus configuration based on tag or address?

2.8 Does diagnostics improve system availability?

2.9 Can fieldbus technology be used in safety-related applications?

BIBLIOGRAPHY

1. *Application Guide, HCF LIT 34*, HART Communication Foundation, 1999.

2. *Draft PROFIBUS Profile, Fail Safe with PROFIBUS*, Revision 1.0, Order No. 3.092, April 1999.

3. *Technical Description, Order Number 4.002,* PROFIBUS Nutzenorganisation e.V., September 1999.

4. *Technical Overview, FD-043,* Revision 2.0, Fieldbus Foundation, 1998.

3

Installation and Commissioning

As explained in chapter 2, the installation and commissioning of field devices is a process greatly simplified by fieldbus technologies. HART, FOUNDATION Fieldbus H1, and PROFIBUS PA were purposely designed to make it possible to use existing wires. They, therefore, do not have as many special requirements for knowledge and tools as do other networks. The electrical installation of FOUNDATION Fieldbus H1 and PROFIBUS PA devices is identical since they are based on the same standard, IEC 61158-2. However, the installation of HART devices is different in several aspects.

Before installing any device make sure you carefully read its manual and follow the manufacturer's instructions and local regulations.

HART

Devices that are a hybrid of the analog 4-20 mA and digital communications are called "smart." The communication speed is low, and normal instrument-grade cable can therefore be used. Its possible to replace analog devices with smart devices without changing the wires. The HART communication is modulated as an AC signal superimposed onto the 4-20 mA DC signal. The communication signal is symmetric and therefore does not affect the 4-20 mA signal. The digital communication and analog signal can hence be used simultaneously, which makes it possible to do configuration, diagnostics, checks and the like while the device is operating in the control loop. The HART communication signal is in a frequency range that is filtered out by most analog devices. This allows them to operate normally undisturbed by the HART signal. A device with 4-20 mA input or output can be connected in the loop as usual.

Topology

For the HART communications protocol there are two possible topologies. The most common by far is point-to-point wiring that in a majority of cases is a traditional two-wire 4-20 mA DC current loop. A regular stabilized DC power supply can be used. The ripple limit can be met by basically any modern regulated power supply. The resistance in the current loop, which is often provided by input shunt resistors, must be 230 ohm or greater for the HART signaling to work. Up to two masters can be connected to a HART network at the same time. The primary master could be a permanently connected interface such as a remote terminal unit (RTU) or a multiplexer for the host. The secondary master is the most common tool and is generally a portable handheld terminal that is only connected temporarily when working in the field. In order for the interface to work it must be connected between the field device and the resistor. If the interface is connected between the power supply and the resistor there will be no communication because the signal is short-circuited by the power supply.

Point-to-Point

In the point-to-point topology only a single field device slave is connected on each pair of wires. A transmitter may be loop powered or separately powered, but the connection is the same as that of a conventional analog transmitter. Many times, a 230-ohm resistor need not be added to the current loop since many recorders, indicators, single-loop controller, and I/O-subsystem input modules already have a 250-ohm shunt resistor on their input (figure 3-1).

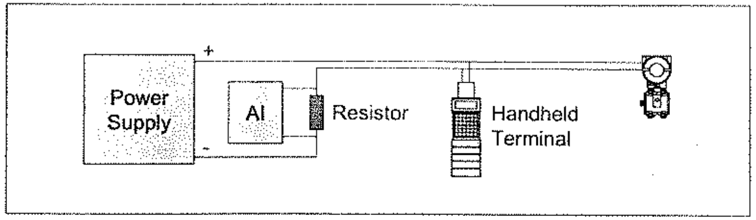


Figure 3-1. HART transmitter connection.

In point-to-point topology, the analog devices in the current loop, such as input modules, indicators, and recorders, are wired in series, which causes a voltage drop over each. Technicians must ensure that the power supply is sufficient for the total resistance of cable and the input shunts otherwise there will not be sufficient operating voltage for the field device when current is at its maximum.

EXAMPLE 3-1

A transmitter may need a minimum of 12 V to operate, and a regular 24 VDC power supply is used. Assuming that the current in the loop is at most 21 mA, then the maximum allowed loop resistance is 571 ohm:

$$\frac{24-12}{21\times 10^{-3}} = 571\ ohm$$

Given that one input shunt generally is 250 ohm, the combined resistance of cable and other devices shall not exceed 321 ohm.

Output devices such as positioners and electric actuators in point-to-point topology are connected to the single-loop controller or I/O-subsystem output module just like their analog counterparts. For most output devices no resistor is required since the field device itself has a sufficiently high resistance to satisfy the HART requirement (figure 3-2). The 4-20 mA output must be able to drive a high enough load to overcome the voltage drop over the field device. The voltage drop across smart positioners is generally higher than for conventional analog positioners. In split-range applications where two positioners are connected a distributor may be required to generate two separate signals to drive the individual devices.

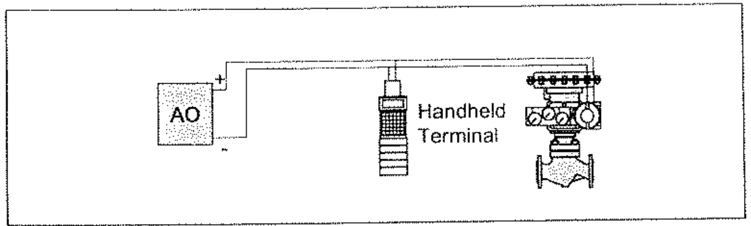


Figure 3-2. HART output device connection (without additional resistor)

EXAMPLE 3-2

A positioner may need a minimum of 11 V to operate and 540 m of cable to be connected. Given that the current in the loop is at most 21 mA and that there is a cable that has a resistance of 22 ohm/km in each of the two conductors, then the current output must be able to swing up to 11.5 V:

$$U = R\times I = 2\times 22\times 0.54\times 21\times 10^{-3} + 11 = 11.5V$$

11.5 V is equivalent of driving a resistive load of 575 ohm.

HART devices may have built-in controller functionality. Independent of the location of the control function the transmitter and positioners are connected in series (figure 3-3). If the control function is in the transmitter the loop current is the manipulated variable. If the controller is located in the positioner the loop current is the process variable. Care must be taken to ensure that a sufficient power supply is available to power both devices and to overcome the loop resistance.

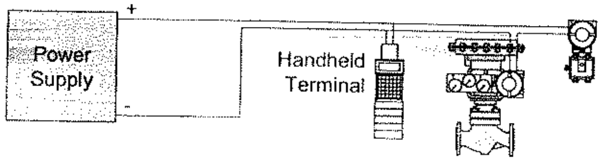


Figure 3-3. HART field device controller connection.

EXAMPLE 3-3

Based on the same conditions as in the previous examples a minimum of 24 VDC is required:

$$U = R\times I = 2\times 22\times 0.54\times 21\times 10^{-3} + 11 + 12 = 23.5V$$

Multidrop

In the multidrop topology several field devices are connected in parallel on each pair of wires (figure 3-4). The field devices may be loop powered or separately powered. Loop-powered transmitters generally have their output set constant to 4 mA, that is, the output no longer reflects the process variable. Therefore, the current in the wire carries no useful information.

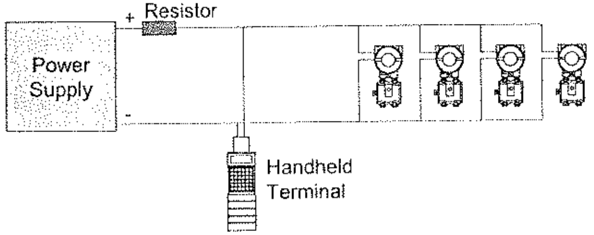


Figure 3-4. HART multidrop mode.

If many devices are connected on a single pair of wires, then the voltage drop over the resistor will be high, and consequently the power dissipation as well. The power supply output voltage and the resistor power rating must therefore be calculated. Typically, no more than fifteen devices are connected on a HART network.

EXAMPLE 3-4

Assume fifteen devices, each consuming 4 mA and requiring 12 V to operate and using 1000 m of cable that has a resistance of 22 ohm/km in each conductor. In this case, the power supply voltage has to be a minimum of 30 V.

$$U = R\times I = 15\times 4\times 10^{-3}\times(250 + 2\times 22) + 12 = 29.6V$$

The power dissipated in the resistor will be 0.9W:

$$P = I^2\times R = (15\times 4\times 10^{-3})^2\times 250 = 0.9W$$

A resistor with a power rating greatly exceeding 0.9W should therefore be used.

When multidropped HART devices are used in controller mode, the 4-20 mA analog signal for every transmitter remains active.

This results in a large total current for a network with many devices. The voltage drop over a resistor would be too great. Therefore, HART power supply impedance should be used instead of a resistor for multidropped controlling devices. Because the power supply impedance has a very small resistance the DC voltage drop is small.

An important consideration for HART multidrop networks is the scan time. Because it takes about a half second to one second to get a reading from each field device, depending on the master and the slaves, it may take as much as 7.5 to 15 seconds to get the reading from fifteen multidropped transmitters. Although this may be sufficient for monitoring many processes, it is not fast enough for most control loops.

Cable

HART devices are able to work with the type of cables commonly used for instrument installation. Because of the electrical characteristics of cables the HART signal gets attenuated as it travels down the wires until it is too weak to be picked up. There is therefore a limit to the cable length (table 3-1).

Table 3-1. Theoretical HART communication limits.

Pair	Shield	Twisted	Size	Length
Multi	Yes	Yes	0.2 mm2 (AWG 24)	1,500 m (5,000 ft)
Single	Yes	Yes	0.5 mm2 (AWG 20)	3,000 m (10,000 ft)

The main limiting factor for the cable length is the capacitance of the cable itself and that of the connected devices (table 3-2).

Table 3-2. Maximum cable length as a function of device quantity and cable capacitance.

Devices	65 nf/km (20 pf/ft)	95 nf/km (30 pf/ft)	160 nf/km (50 pf/ft)	225 nf/km (70 pf/ft)
1	2,800 m (9,000 ft)	2,000 m (6,500 ft)	1,300 m (4,200 ft)	1,000 m (3,200 ft)
5	2,500 m (8,000 ft)	1,800 m (5,900 ft)	1,100 m (3,700 ft)	900 m (2,900 ft)
10	2,200 m (7,000 ft)	1,600 m (5,200 ft)	1,000 m (3,300 ft)	800 m (2,500 ft)
15	1,800 m (6,000 ft)	1,400 m (4,600 ft)	900 m (2,900 ft)	700 m (2,300 ft)

Intrinsic safety put additional limitations on these lengths, in some cases making the maximum distance shorter. The cable shield shall only be grounded in one point, generally at the I/O subsystem.

Intrinsic Safety

Intrinsic safety is a method for preventing ignition caused by electrical faults by limiting the power available to the bus wires in the hazardous area (figure 3-5). This allows devices to be connected and disconnected under power. The basic concept of intrinsic safety and area classification is beyond the scope of this book, but *Electrical Instruments in Hazardous Locations* (fourth edition, E.C. Magison, 1998, 1-55617-638-4) is an excellent reference on the subject. Most of the intrinsic safety rules and principles that apply for all systems, such as grounding, also apply to networks.

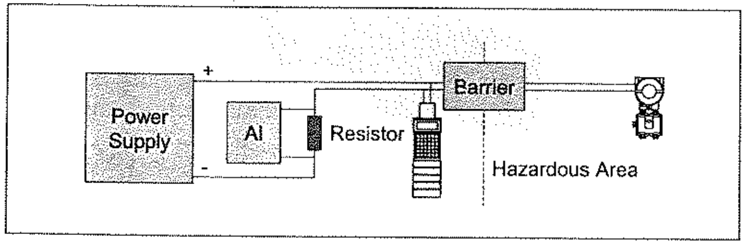


Figure 3-5. Intrinsically safe HART transmitter connection.

To enable bidirectional communication with the field device make sure to use zener or isolating barriers that are HART compatible. Care should be taken when using zener barriers because the voltage drop over the internal resistance limits the voltage available to the field device and the rest of the current loop. Isolating barriers make a higher voltage available to the field device, which ensures that there is sufficient voltage to the device and makes it possible to use longer cables. Different barriers are required for transmitters and positioners.

HART interfaces and handheld terminals are not passive devices. They add voltage to the bus, typically a maximum of 2 V. Therefore, the allowable capacitance should be reduced by 15 percent. Any device connected on the hazardous-area side of the barrier must be intrinsic safety approved.

WARNING—For intrinsic safety design and installation, follow local regulations and codes of practice and make sure to read the respective user manuals. Connect only intrinsically safe certified devices in the hazardous area.

Host and Portable

In most installations communication with HART devices is achieved by using a temporarily connected portable handheld terminal. A portable computer can connect to the network using a HART interface. The portable can be connected at any access point along the wires, for example, at the marshalling panel or a field junction box. Generally, field devices have connection points on the terminal block to make connection easy. For a HART-enabled control system the host is permanently connected to the field-level networks. This is done either directly with an interface or via an I/O subsystem such as a multiplexer or conventional I/O modules that have the ability to let HART communications pass through. A multiplexer generally has many communication ports, thus supporting several HART networks.

Handheld Terminal

The handheld is generally a secondary master that is temporarily connected to the network. The handheld may be a dedicated HART communicator with integral interface or a commercial off-the-shelf organizer with a plug-in HART interface and software (figure 3-6).

Interface

Generally, the computer interfaces plug into the RS232 serial communication port (figure 3-7). This makes it possible to use a notebook PC for maintenance work. The larger screen makes it easier to monitor the device data, eliminating several levels of menus.

Multiplexer

A multiplexer has several HART communication ports, each connected point to point or in some cases multidrop to the field devices. The multiplexer is typically a primary master that is permanently connected to the networks. The multiplexer continuously scans the process variables from all transmitters, independently of the host, and at the same time receives basic device status information. The host in turn reads the process variables from the multiplexer.

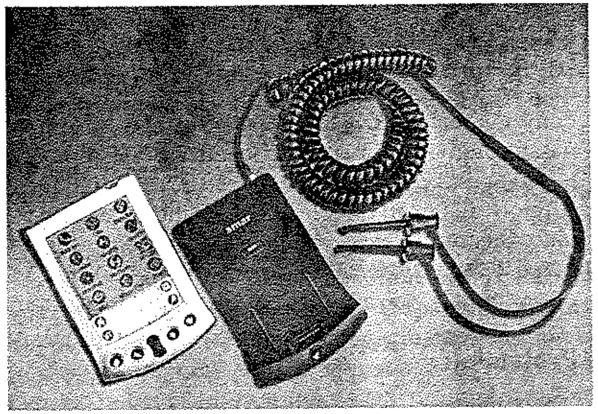


Figure 3-6. Portable configuration tool for HART devices. (Courtesy of Smar)

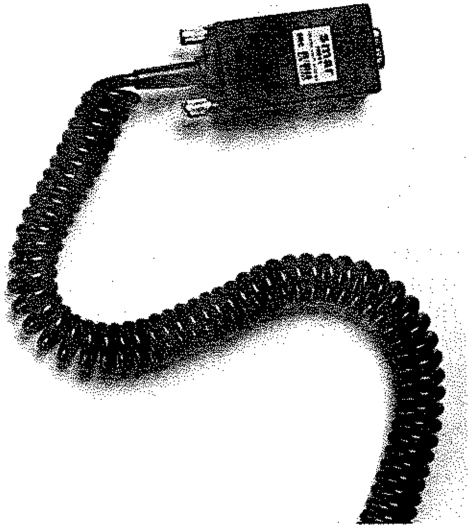


Figure 3-7. HART interface for serial port. (Courtesy of Smar)

An important feature is passthrough capability allowing for direct communication between the host and the field devices. The data from the HART networks are multiplexed onto a single network, typically Modbus, PROFIBUS DP, or a proprietary protocol running on RS485 to the host computer. Thus, a multiplexer is essentially a gateway. With a permanent connection and data passthrough it is possible to benefit from asset management features such as continuous diagnostics, provided the host software supports it. On an existing system where the field I/O is already in place the multiplexer may be installed in parallel with the conventional I/O subsystem. For new installations using HART the multiplexer may be an integral part of the I/O subsystem.

Surge Protection

HART devices are generally designed to operate on voltages up to 45 VDC. This is in about the same range as for FOUNDATION Fieldbus H1 and PROFIBUS PA because the IEC 61158-2 standard specifies a normal range of 9-32 VDC, with a maximum of 35 VDC. Therefore, the surge protection methods and equipment for all three buses are very similar. A surge is a voltage that is much greater than the normal operating voltage, typically only present for a short time, and would damage unprotected electrical equipment. A direct lightning strike, lightning strike to the ground nearby, or switching of inductive loads may cause the surge. The surge voltage can exceed tens of kilovolts and last only for a fraction of a second. Damage can be prevented using surge-protection devices. Direct lightning strikes are rare, but strikes to the ground are common in some areas. They cause currents of tens of kilo amps to flow through the soil, which in turn cause large ground potential shifts between points separated by some distance. Although this does not present any problem within a confined location such as a building, the electrical earth potential difference between two locations can lead to currents of thousands of amps flowing through and destroying a network connecting the two points. Surge protection should not be applied only to the network, but also to the power supply, telephone lines, and the like. A surge protection device has to be installed at every field instrument as well as at the marshalling panel where the field wiring enters to the host (figure 3-8).

The surge protection device is normally deactivated, but when a surge exceeds the threshold voltage the internal circuit, such as a gas discharge tube, diode, and varistors, channels the current to earth instead of through the device. Therefore, a good earth connection is very important. The surge protection device for field instruments mounts directly onto the housing in a spare electrical conduit connection (figure 3-9). If this is not available it should be installed in a junction box no further than one meter away.

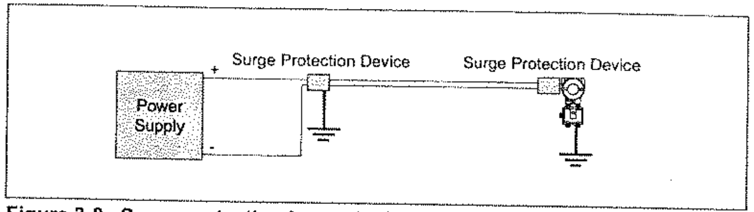


Figure 3-8. Surge protection has to be installed in both ends of the network.

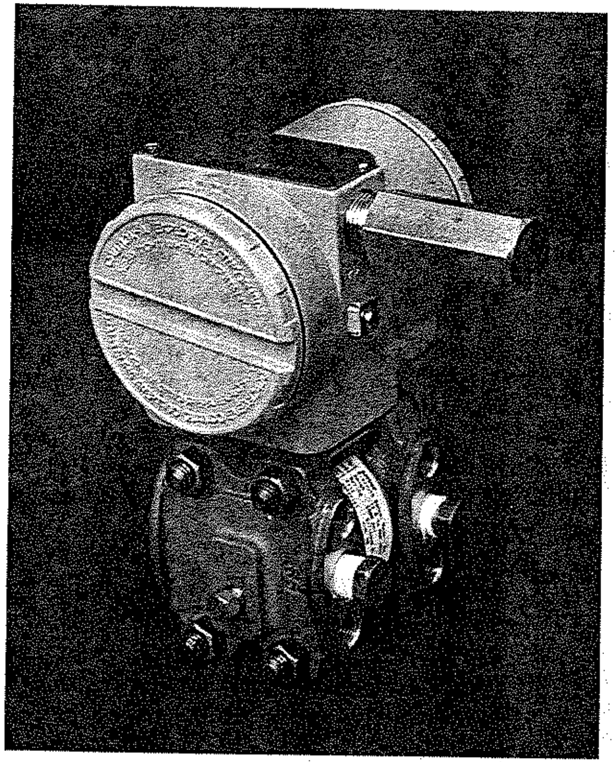


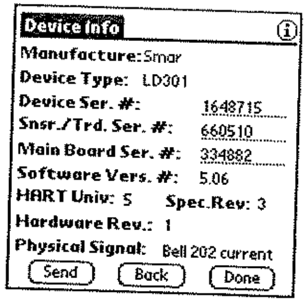
Figure 3-9. Field device mounted surge protector. (Courtesy of Smar)

Good surge protection devices survive repeated surges and let the protected devices resume normal operation once the surge is over. The host-end surge protection device is typically mounted on a DIN rail in the marshalling cabinet.

Device Commissioning

HART devices have two addresses, a unique hardware address and a polling address. The unique hardware address consists of a manufacturer code, a device type code, and a unique identifier. This ensures that no two addresses in the world are the same, eliminating the chance of conflict. Every device is shipped from the factory with the hardware address set. It cannot be changed and can therefore never be mixed up. The hardware address is normally not shown to the user. For point-to-point mode the polling address is set to zero, in which case the 4-20 mA output is still active for transmitters. For multidrop mode the polling address is set in the range of 1 to 15, which in the case of transmitter mode sets the output fixed at 4 mA. If multidropping is used, every field device should be configured with a different polling address before it is connected to the network. This is typically done by first connecting the device point to point and manually assigning the new address from the handheld terminal. If the polling address is duplicated communication cannot be established.

The digital communication makes commissioning a field device a much faster process. Once installed, device information can be verified from the control room to ensure that the correct device has been connected to the wire (figure 3-10).



Manufacture: Smar
Device Type: LD301
Device Ser. #: 1648715
Snsr./Trd. Ser. #: 660510
Main Board Ser. #: 334882
Software Vers. #: 5.06
HART Univ: 5 Spec.Rev: 3
Hardware Rev.: 1
Physical Signal: Bell 202 current

Figure 3-10. Remotely accessible device identification simplifies verification. (Courtesy of Smar)

Verification can be taken further using the loop-test feature in transmitters. Through the handheld terminal, the transmitter is set to generate a specific output current independent of the applied input in order to simulate any particular process condition. It is therefore possible to test out the loop integrity and control strategy response for normal and extreme process conditions without changing the process or having to venture into the field with a simulator.

IEC 61158-2 (FOUNDATION Fieldbus H1 and PROFIBUS PA)

Both FOUNDATION Fieldbus and PROFIBUS PA have physical layers based on IEC 61158-2. This standard is purely digital, with no 4-20 mA on the network as in the case of HART. Although they are electrically identical, the two types of devices cannot be mixed on the same network. The electrical installation is the same, but the specifications use slightly different terminology, and the commissioning is done differently as well. Surge protection for FOUNDATION Fieldbus and PROFIBUS PA is performed in the same way as for HART because the operating voltage range is about the same. The field-level network is relatively low speed. This makes installation quite forgiving in comparison with many other protocols. The rules for topologies, cable, distances, device quantities, and the like are not very strict.

Terminology

A network may consist of one or more segments, but in most installations only a single segment per network is used (figure 3-11). Multiple segments are connected using repeaters or other devices containing a repeater, such as a safety barrier. A segment must have two terminators, one in each end. Thus, a network with multiple segments has several terminators in total. Often, the terminator is built into another device, eliminating the need for a separate terminator. From a software configuration perspective the hardware network is sometimes called a link. Fieldbus devices connect to the network through their communications port. Field instruments typically have only one port, whereas the interface for the host typically has several ports in a modular fashion.

Multiple-segment networks are common in intrinsic safety applications where safety barriers with built-in repeaters are often used. Repeaters that are multidropped on a safe-side segment join several parallel, intrinsically safe segments to form a network in

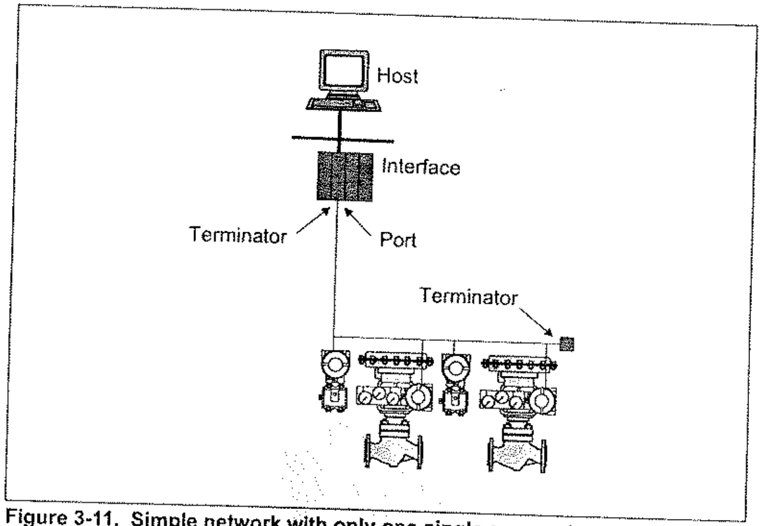


Figure 3-11. Simple network with only one single segment.

which many devices are connected to a single interface port (figure 3-12).

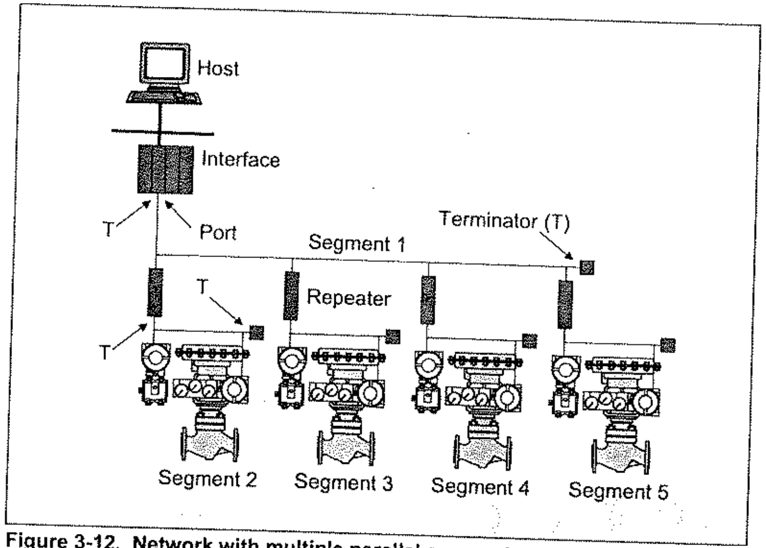


Figure 3-12. Network with multiple parallel segments common for intrinsic safety.

Multiple segments can also be chained in series to form a network that reaches further (figure 3-13).

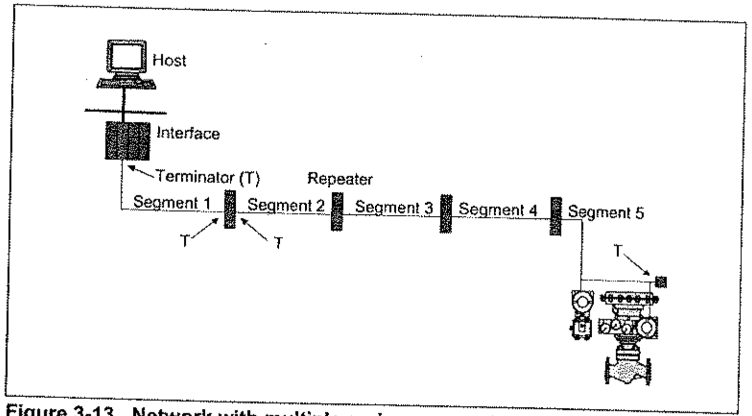


Figure 3-13. Network with multiple series segments used for long distance.

Devices on the field-level network may either be powered by the network, that is, bus powered, or powered separately. Some devices may be both bus powered and separately powered, as in the case of many electrically actuated valves where the positioner typically is bus powered while the actuator is separately powered. The field-level network devices are powered either by a regular power supply through an impedance or, in the case of intrinsically safe installations, through a safety barrier.

A fieldbus network has two wires used for both communication and power supply to bus-powered devices. All network devices are connected in parallel (figure 3-14). It is possible to connect or remove a device while the bus is running. Each segment of the fieldbus network has to have two terminators, one in each end. If bus-powered devices are connected to the segment power must be supplied to the network. Depending on whether or not intrinsic safety is required the power supply subsystem must be different. Power supply, host-end terminator, and host interface may in some cases have been integrated into a single device to simplify wiring in the panel.

The terminator (figure 3-14) is a simple component that does not communicate. One terminator is connected in each end of the network. The terminator has two functions, the first being the classical terminator function of preventing signal reflection and subsequent communication errors. A device transmits by rapidly changing the

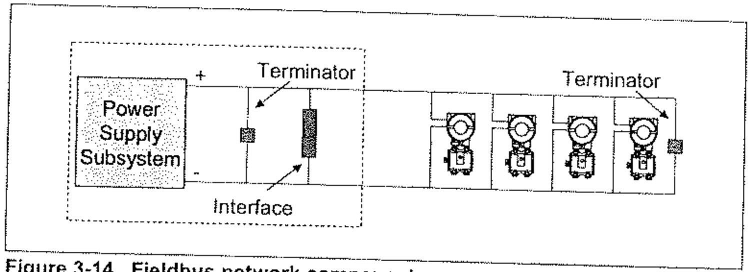


Figure 3-14. Fieldbus network components.

current in the network, either by changing its power consumption or injecting a current. The second function of the terminator is to convert the current change produced by a transmitting device into a voltage change across the entire network that is picked up by all devices as the means for receiving the signal. The terminator only has two terminals and is polarity insensitive (figure 3-15). On a test bench, a fieldbus network may function with only one terminator, but in a real installation with long wires it will not work reliably. Because there are different terminators for each type of network, it is important to use a terminator designed with the characteristics stipulated in IEC 61158-2 and not some other kind of bus.

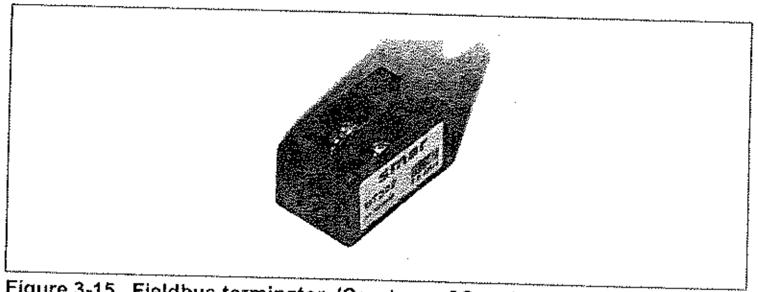


Figure 3-15. Fieldbus terminator. (Courtesy of Smar)

Topologies

The field-level network can be laid out in two basic topologies, bus topology and tree topology. An important difference between the two topologies is where the terminator shall be mounted. For bus topology, a main cable called "trunk" typically runs from the marshalling panel near the central control room out into the field until it reaches the last device, passing all the devices along the way. From the main trunk shorter wires called "spurs" connect the individual devices (figure 3-16). For bus topology, it is easy to identify which is the longest cable: the trunk. The two terminators are mounted at the ends of this trunk. Because the power-supply end of the network typically has a built-in terminator, only the field-end terminator has to be purchased as an individual part. A segment is not a spur. Every segment of a network is using a repeater, and each can have several spurs.

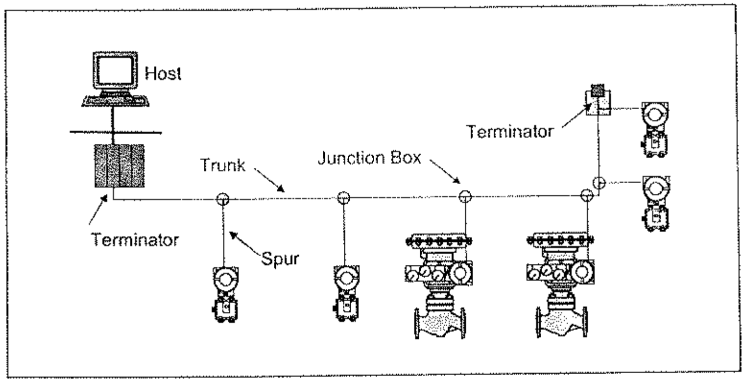


Figure 3-16. Bus topology with terminator at end of the trunk at last device.

The spurs may connect to the trunk either by using small junction boxes (figure 3-17) or by using molded "T" connectors. The advantage of the bus topology is that it minimizes the amount of cable that is required to connect the devices. However, because several junction boxes are required for the spurs there will be many connections. The terminator is typically mounted in the furthest junction box along the trunk.

For tree topology, the trunk runs into the field where it branches out into the individual devices from a single junction box (figure 3-18). Since the trunk ends in the junction box the field-end terminator is mounted in the junction box. This topology minimizes the number of connections, but the cable length will be longer, especially if the devices are not particularly close together. Tree topology is common when older plants are reinstrumented while retaining the existing wiring.

Devices can be installed and removed under power. However, care must be taken to ensure that the continuity of the trunk circuit is maintained and that loose leads don't touch each other and short-circuit the network. This would mean a loss of communication and controls for the devices on the segment.

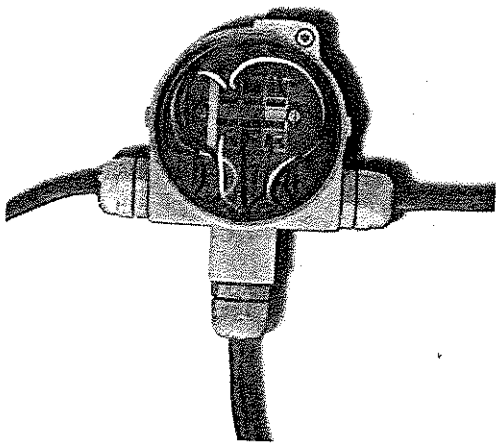


Figure 3-17. Junction box suitable for tapping spurs off the trunk. (Courtesy of Smar)

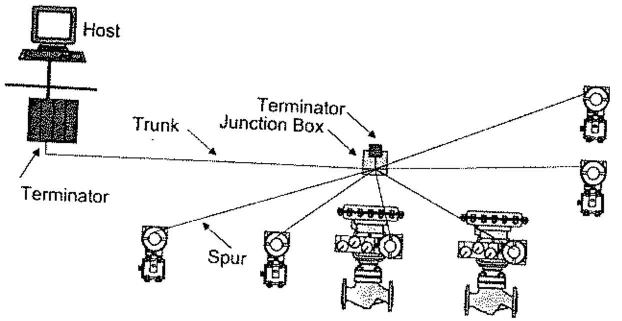


Figure 3-18. Tree topology with field-end terminator where trunk ends and spurs branch out.

It may be helpful to use junction boxes with plug-in connectors. This will facilitate the quick connection and disconnection of device wiring without the risk of disrupting the bus (figure 3-19).

Active junction boxes contain current-sensing electronics that detect if the current on a spur branching out to one of the devices exceeds a set limit, typically 30 mA. If it does, this is interpreted as meaning that the spur has been short-circuited. In this case, the spur is isolated from the rest of the bus to prevent prolonged disruption of the network. Active junction boxes can be used to ensure that not all of the devices on the network will be affected if a short-circuit caused by improper handling of the wires occurs. Quick-connect junction boxes are typically industrially hardened and weatherproof, so there is no need to mount it inside another junction box. This eliminates the need for cable glands and terminals.

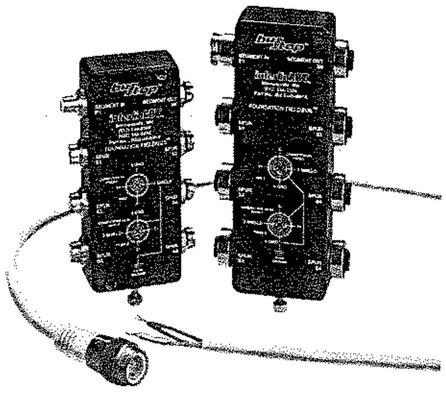


Figure 3-19. Weather proof quick-connect junction box. (Courtesy of Turck)

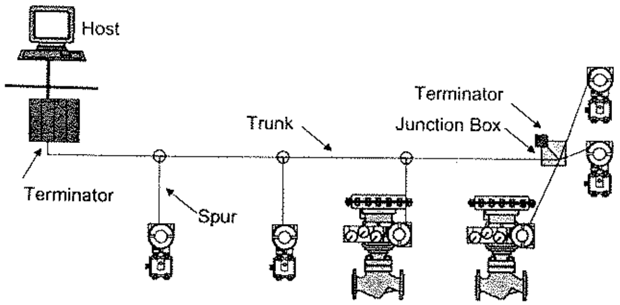


Figure 3-20. Mix of bus and tree topologies.

Depending on the geographical layout of the plant it may be advantageous to mix tree topology and bus topology to achieve optimum wiring (figure 3-20). The IEC 61158-2 network does not support ring topology.

It is important to connect devices in such a way that one can easily be removed without disrupting the entire bus. That means, for example, that if a device has two conduits, you should not run the trunk in through one conduit connection and out through the other since doing so will mean that the circuit must be opened to remove the device.

Because the input impedance of each field device acts as a load on the communication signal, a maximum of 32 devices can be connected per network if all of the devices are separately powered. However, most fieldbus devices are bus powered, and the voltage drop along the conductors limits the number of devices to more like 16 for a single segment network. For a larger network that is made from several segments joined by repeaters, the 32-device limit applies per segment, and the network may therefore theoretically have as many as 240 devices for FOUNDATION Fieldbus and 126 for PROFIBUS. Wire resistance, current consumption, interface memory, and update time impose restrictions that keep the realistic number around 16 for the whole network. Plants therefore do not have just one field-level network but many. Bridges interconnect the many networks throughout the plant, allowing dispersed devices to communicate with each other.

Do not connect devices to the bus that have not been designed according to IEC 61158-2. Such devices usually disrupt the communication signal.

Cable

IEC 61158-2 does not specify the type of cable or characteristics that should be used for installations. However, the cable specified for conformance testing purposes can be taken as a recommendation. The standard was specifically designed to be able to work with the type of cables commonly used for instrument installation in order to make migration of existing installation simple. The cable type and characteristics are major factors determining the bus length and number of devices that can be used. The rules for device quantity and cable length are not absolute, and many are based on worst-case conditions. Most of the time the rules can be bent and interpolated using good judgment.

Cable Type

Because of the electrical characteristics of cables, the fieldbus signal gets attenuated as it travels down the wires until it is too weak to be picked up. Consequently, there is a limit to the length of the cable. In an annex included for informational purposes, IEC 61158-2 states the typical expected total length limit for four types of cable (table 3-3). Cable similar to example A or B is recommended for new installations. The type B multi-core cable is well suited for use as a "homerun" cable from the field junction box to the marshalling panel near the control room. Note that twisted-pair and shielding is critical to gain immunity to interference and to enable use across long distances. Other cable types than those given in table 3-3 may be used, but the table serves as a good guideline to the expected maximum length that can be achieved. Cables made from polyolefin materials generally have more stable electrical characteristics such as impedance than do, for example, PVC.

Table 3-3. Typical cable types and range.

Pair	Shield	Twisted	Size	Length	Type
Single	Yes	Yes	0.8 mm2 (AWG 18)	1,900 m (6,200 ft)	A
Multi	Yes	Yes	0.32 mm2 (AWG 22)	1,200 m (3,900 ft)	B
Multi	No	Yes	0.13 mm2 (AWG 26)	400 m (1,300 ft)	C
Multi	Yes	No	1.25 mm2 (AWG 16)	200 m (650 ft)	D

Device power consumption and intrinsic safety put additional limitations on these lengths, in some cases making the maximum distance shorter. However, the maximum total length of the cable depends on the cable characteristics. As a result, 1.9 km is not a limit. Other types of cable better than type A may be used for even longer distances. Cable types in a network can be mixed. For example, in a tree topology a multi-core cable may be used for the homerun cable from the marshalling panel to a field junction box, from where single-pair cables branch out to the individual devices.

Because fieldbus was designed to work with cable found in existing plants most common instrumentation wire meets the fieldbus specification. However, for new installations it is recommended that you use cable that meets or exceeds type A and B.

Length

The total length summing the length of the trunk and that of all the spurs must not exceed the limitation for the particular cable type, for example 1.9 km in the case of type A cable. For longer distances, a network may reach farther that is made from several segments joined by repeaters because the cable limit applies per segment. The shorter the total cable length the better, so avoid unnecessarily long cable routing. For most of the distance, the main trunk typically is a multi-core homerun cable that is shared by many networks from a field junction box into the marshalling panel.

The total cable length for the network example in figure 3-21 is calculated in table 3-4.

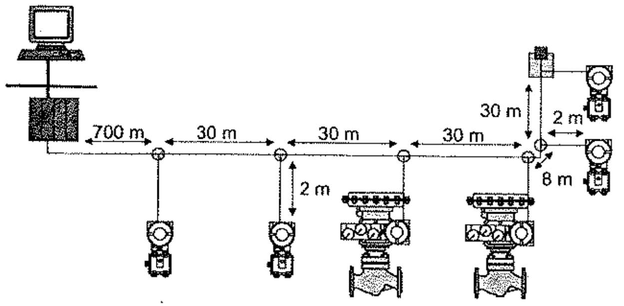


Figure 3-21. Example of a network having 840 m total length.

Table 3-4. Cable length calculation for the network in the figure 3-21 example.

Trunk	700 m (2300 ft)
Trunk	30 m (100 ft)
Trunk	30 m (100 ft)
Trunk	30 m (100 ft)
Trunk	8 m (24 ft)
Trunk	30 m (100 ft)
Spur 1	2 m (6 ft)
Spur 2	2 m (6 ft)
Spur 3	2 m (6 ft)
Spur 4	2 m (6 ft)
Spur 5	2 m (6 ft)
Spur 6	2 m (6 ft)
Total	840 m (2760 ft)

Unless the distances at the site are very large because the control room is located far from the field or the devices are widely spread out, there may be no need to make cable-length calculations for a majority of the networks. For most installations, the total distance of any network is far less than the maximum allowed for the type of cable used, and length is thus not a limiting factor. Perhaps only in some suspect, borderline cases where wire has to run farther to a remote plant unit will calculation be required.

Keep the parts of the cables that have no shielding, or where the conductors are not twisted, below 2 percent of total cable length or 8 m, whichever is shorter.

When bus-powered devices are used, which is almost always the case, the voltage drop along the wire caused by the current consumption of the field devices also limits wire length. For maximum range and number of devices the supply voltage shall be as high as possible, the wire cross-section as large as possible to reduce resistance, and the field device current consumption as low as possible. The maximum distance can be calculated using Ohm's law.

If the power supply output voltage is lower, or the device power consumption is higher, the distance will be shorter. Conversely, the opposite also applies. It is therefore critical for both intrinsically safe and regular installations that the device current consumption be as low as possible. Even many devices that receive separate power still draw some current from the fieldbus network.

Low power consumption is important for devices that are to be used in intrinsically safe networks.

Spurs

Spurs are short pieces of cable that branch out from the main trunk to each individual device. A cable that is shorter than one meter is not counted as a spur but simply considered a splice. Spurs are a good way to connect devices to the trunk because they make it simpler to remove a device for maintenance without disturbing the rest of the network. There are limits to the number of spurs that can be on a network and to the length of a spur. Table 3-5 provides a guideline to the number of devices that can be put on each spur and to the maximum spur length. Spur length is basically independent of the type of cable that is used but depends on the number of devices connected to it and the whole network. There should never be more than four devices connected to each spur.

EXAMPLE 3-5

As defined in the standard, a fieldbus device needs a minimum of 9 V to operate. For nonintrinsically safe networks the power supply voltage typically exceeds 21 V, and current consumption for a typical device is 12 mA. Type A cable has a resistance of 22 ohm/km in each of the two conductors. For a network populated with sixteen devices the maximum distance is calculated as 1.4 km:

$$L = \frac{U}{I \times r} = \frac{21-9}{12 \times 10^{-3} \times 16 \times 2 \times 22} = 1.4$$

For 1.9 km of type A cable the maximum number of devices is calculated as twelve:

$$n = \frac{U}{I \times R} = \frac{21-9}{12 \times 10^{-3} \times 1.9 \times 2 \times 22} = 12$$

For an intrinsically safe installation with eight devices using a safety barrier with 13.8 V output, the maximum distance is calculated as 1.1 km, considering only the voltage drop. However, other aspects set the limit for intrinsic safety (Exia) gas group IIC as 1 km:

$$L = \frac{U}{I \times r} = \frac{13.8-9}{12 \times 10^{-3} \times 8 \times 2 \times 22} = 1.1$$

Table 3-5. Spur length as a function of number of spurs and number of devices per spur.

Number of Spurs	1 Device	2 Devices	3 Devices	4 Devices
25-32	–	–	–	–
19-24	30 m (100 ft)	–	–	–
15-18	60 m (200 ft)	30 m (100 ft)	–	–
13-14	90 m (300 ft)	60 m (200 ft)	30 m (100 ft)	–
1-12	120 m (400 ft)	90 m (300 ft)	60 m (200 ft)	30 m (100 ft)

Particular attention should be paid to spur length with regard to tree topology since every branch is a spur, and each branch may be quite long. The junction box should be mounted centrally among the devices to avoid wire runs that exceed the limit for spurs (figure 3-22). If any spur is longer than 120 m the terminator should be moved to make the spur part of the trunk.

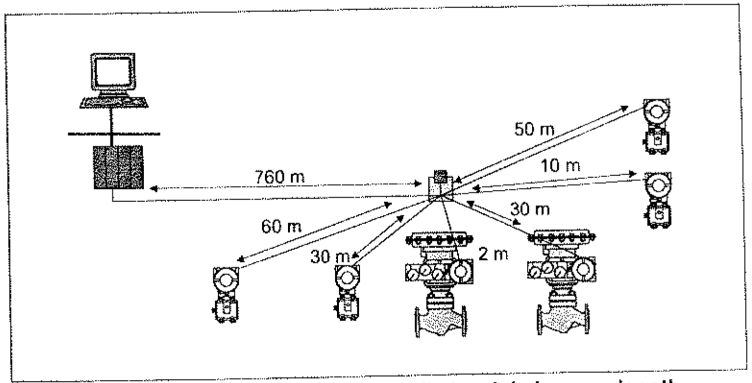


Figure 3-22. Junction box mounted centrally to minimize spur length.

For intrinsically safe installations the spur length should not exceed 30 m.

Connectors

No special connectors or terminal blocks are needed to join spurs to the trunk or connect to devices. Most fieldbus devices have screw terminal connection. However, the use of plug-in connectors makes it easier to safely connect and disconnect devices during operation (figure 3-23). Spur cables with molded connectors reduce wire terminators and are therefore a good option if the required length is known reasonably well at the time of engineering.

Wiring

The common rules for routing signal cables also apply to fieldbus. Keep any signal wires separate from sources of noise. Fieldbus signals and non-fieldbus signals should not be allowed to coexist within the same multi-core cable because noise may be introduced. This is particularly true for switching of heavy loads and drives, but even some older communication signals will cause noise as well. If you must have the wires cross, make sure they only do so at right angles. The fieldbus signal is polarized, but most devices have circuitry that detects and corrects signal polarity. However, many bus-powered devices require correct polarity because of the power supply.

On the network all devices are connected in parallel, that is, all negative terminals together and all positive terminals together. Preferably, you should use color-coded and properly labeled wires to distinguish positive from negative and to identify pairs in multi-core cables. Trunk cables should also be properly labeled in the

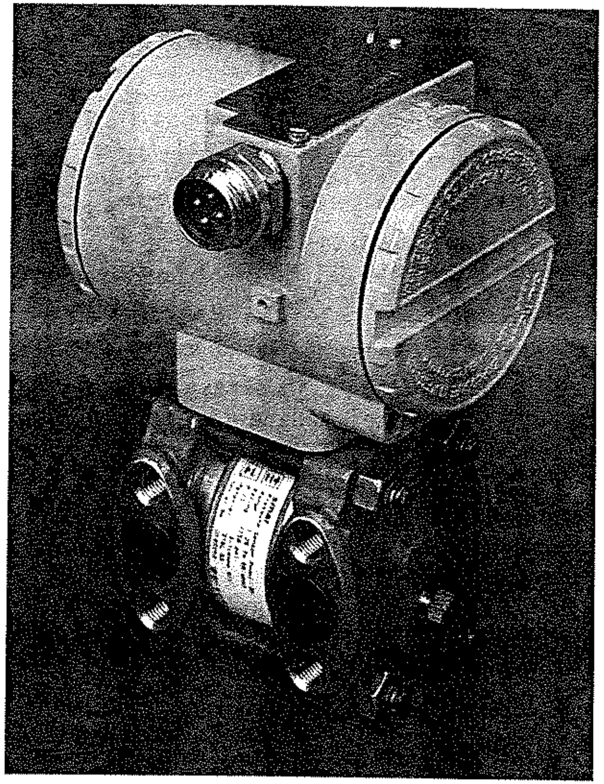


Figure 3-23. Plug-in type quick-connection for field instrument wiring. (Courtesy of Smar)

marshalling rack to reduce the risk that the wrong cable will be disconnected during maintenance. It is a good idea to enter the cable name in the engineering and maintenance software so there is an easy cross-reference between the device diagnostics and the cable.

All kinds of devices can be mixed on the same network, analog and discrete, input and output, as long as they use the same protocol. To reduce the number of loops affected by a network failure and to improve communication efficiency put devices that are used in the same loop on the same network.

Shielding and Grounding

To provide the best immunity from noise, the network cable should be shielded, but alternatively it may be placed inside a metal conduit. The shield shall cover at least 90 percent of the total wire length. Because plants are typically large and widely distributed geographically ground potential differences usually exist. The cable shield should therefore only be grounded in one end, usually at the negative of the power supply (figure 3-24). In some cases, the shield can also be connected to the protective earth. Specific grounding requirements may apply for intrinsic safety.

If multiple grounds are required for an excessively noisy environment, you should employ capacitive grounding. Capacitive grounding means the connection to ground is made through a capacitor rather than a wire, thus only high frequencies are grounded. The shields of the spurs should be connected to the shield of the trunk. Since only the shield should be connected to earth, don't connect the signal wires to earth, and never use a shield as a conductor. All fieldbus devices have communication ports that are galvanically isolated.

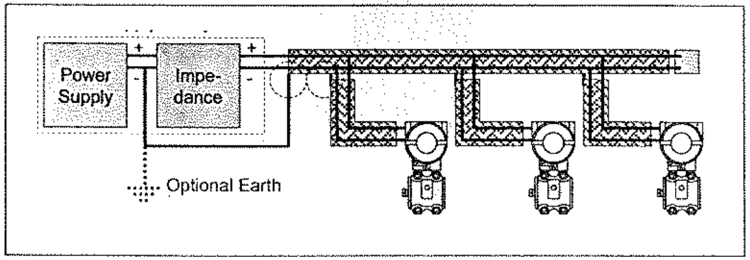


Figure 3-24. Shields of spurs connected to trunk shield and grounded.

It is common for the DC mains power supply to be separate from the power supply impedance module. In this case, the negative of the DC power supply can also be connected to earth if the design of the impedance permits. Grounding either signal conductor may result in the loss of communication. In addition to these operational requirements there may also be safety considerations. The negative of the power supply impedance is a live signal, and it should not be connected to earth. Ideally, the network power supply should be modular with separate DC power and impedance units. The shield can then be grounded at the power supply negative, while at the same time the output of the power supply impedance is free floating in a balanced mode. I.e. the power supply impedance must be balanced or galvanically isolated.

Repeater

Because of the signal attenuation along the wires there is a limit to how long a network segment can be. A repeater contains a chip that refreshes the timing of the signal received in one end and boosts the level as the signal is regenerated in the other end. Thus, the repeater restores the original symmetry and amplitude of the signal, overcoming attenuation. The repeater is bidirectional and typically galvanically isolated. Repeaters may thus be used to connect segments together to form a larger network. As many as 32 repeaters can be connected to the same network segment. However, there may be no more than four repeaters between any two devices on the network, that is, a maximum of five segments separate two devices (figure 3-25). If four repeaters are connected in series, a total distance of 5 times 1.9 km, or 9.5 km, can be achieved using type A cable.

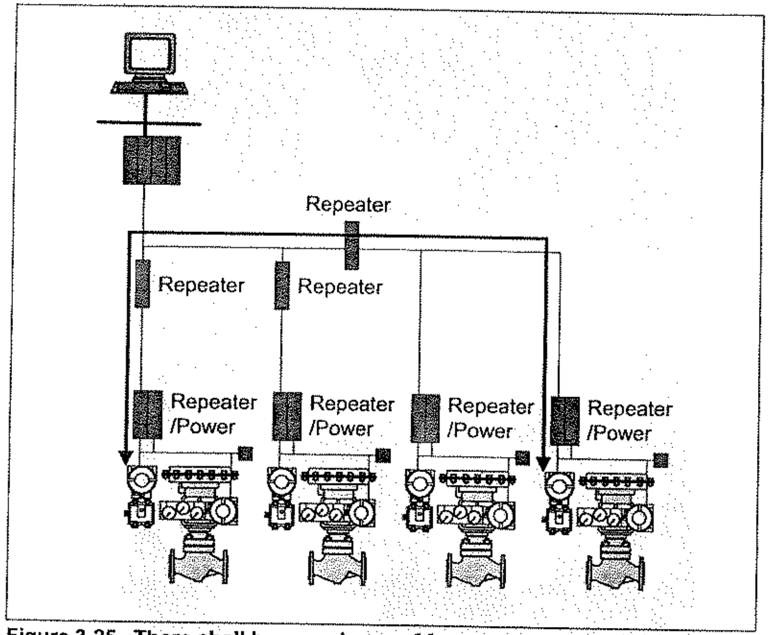


Figure 3-25. There shall be a maximum of four repeaters between any two devices.

Every segment has to have a terminator in each end. However, typically the repeater already has terminators built in. and all that needs to be done is to make sure that the terminators are enabled or disabled as required (figure 3-26). Repeaters do not provide power to the segments, so power must be supplied for those segments that have bus-powered devices. Power must also be supplied on separate terminals for the repeater.

It is helpful if the repeater is isolated and has a terminator built in on both ports.

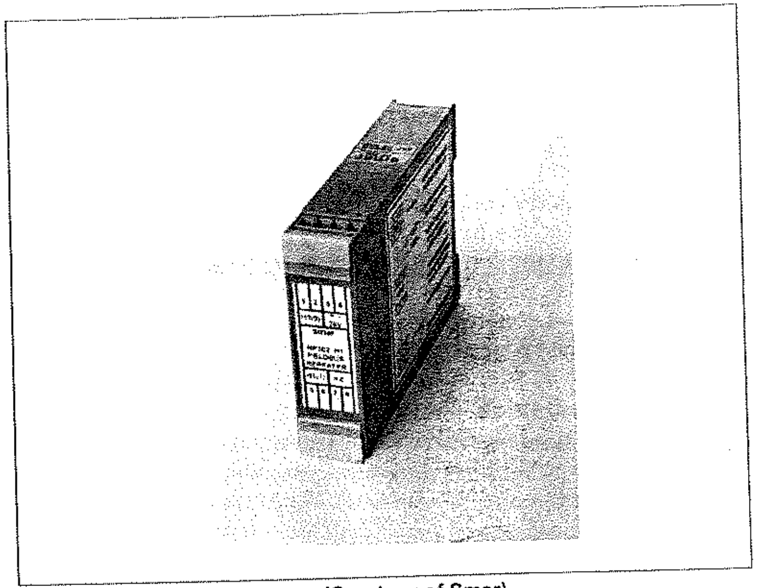


Figure 3-26. Fieldbus repeater. (Courtesy of Smar)

There are different repeaters for each type of network. It is therefore important to use a repeater that is designed with characteristics as stipulated in IEC 61158-2 rather than some other kind of bus.

Host and Portable

The host is connected to the field-level networks either directly with an interface or via a host-level network through a linking device. Portables such as a notebook computer may connect via an interface. The interface or linking device connects to the field-level network through a port. An interface card or linking device may have more than one port, allowing several field-level networks to be connected to a device. Several linking devices can be connected to a host-level network to make very large systems possible. Similarly, interfaces may be designed such that several cards or modules can be plugged into one computer or I/O subsystem.

Typically, there is a limit to the number devices that each port of the interface can support. The wire resistance and device consumption poses an electrical limitation of about sixteen devices for most installations, so interfaces in the market typically only support this number of devices.

Interface

An interface may be a communications module that plugs directly into the conventional I/O subsystem backplane of a traditional system. In PROFIBUS terminology, a PLC is a class 1 master that does cyclic reading and writing of I/O. An interface may also be a card that plugs into a computer. In PROFIBUS terminology, a configuration tool is a class 2 master that does acyclic reading and writing of the configuration. Computer interfaces come in several different styles. Internal ones plug into a slot on the backplane, connect to the computer's USB port, or are inserted as PCMCIA (PC cards), which slot in. For PROFIBUS, the host interface is typically on a DP network, and a signal coupler or link is used to connect to the PA network.

Linking Device

A linking device is used to connect the field-level network directly to the host via the higher-speed host-level network or possibly by way of a remote-I/O network that eliminates the need for I/O-subsystem interface modules. The linking device buffers messages to take care of the difference in transmission speed between the two network levels (figure 3-27).

Linking devices are available for connecting a PROFIBUS PA network to a higher-level PROFIBUS DP (Distributed Peripherals) network and for connecting a FOUNDATION H1 network to an HSE network (figure 3-28). A linking device, fieldbus device power supply, impedance and terminator may be integrated into a single device. A linking device may have multiple ports and may also perform auxiliary tasks, for example, operating like a traditional controller.

Simplify installation by using linking devices that have multiple ports and a built-in field-device power subsystem.

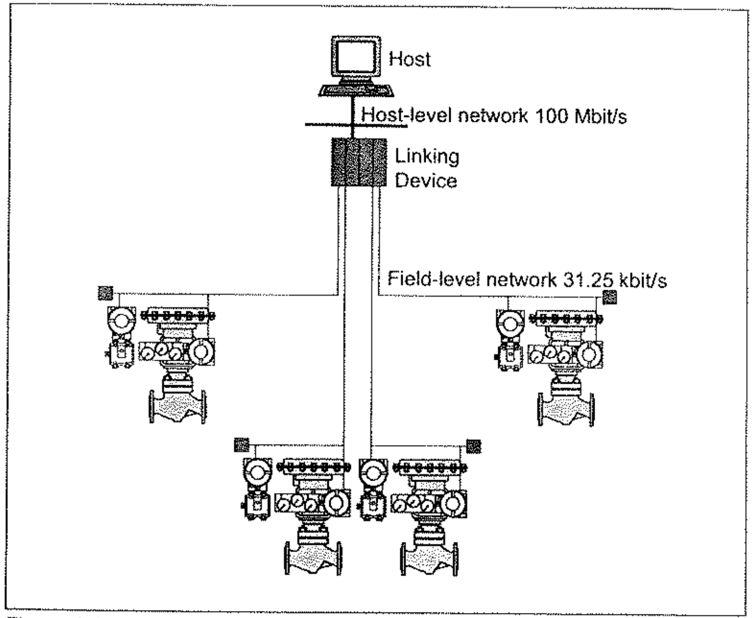


Figure 3-27. Linking device connects field-level with host-level taking care of the difference in speed.

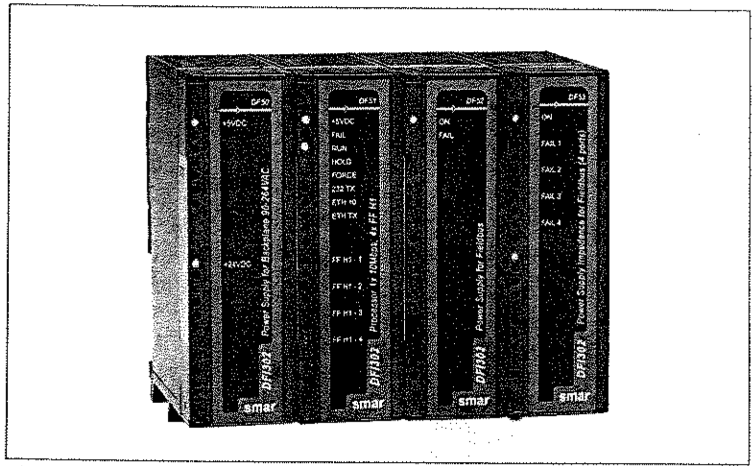


Figure 3-28. Four port FOUNDATION Fieldbus H1 to HSE linking device and H1 to H1 bridge. (Courtesy of Smar)

Where high availability is required you should use redundant linking devices. In other words, two independent linking devices should be connected to the field-level network and two separate data paths should be maintained all the way to the host workstation. This will ensure that plant-floor data reaches the operator even if one interface fails by always providing a window to the process (figure 3-29).

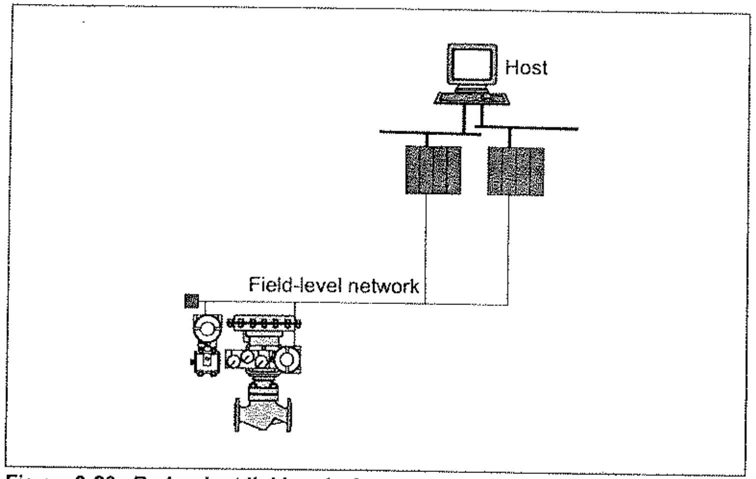


Figure 3-29. Redundant linking devices provide dual paths for data increasing availability.

Coupler

Signal couplers are available so a PROFIBUS PA segment can be connected to a higher-level PROFIBUS DP segment. The difference between a DP/PA coupler and a DP/PA link is that the coupler limits the speed of the DP network to 93.75 kbit/s, whereas the link allows the DP network to operate as high as 12 Mbit/s. In other words, a coupler reduces the performance of the PROFIBUS DP, but a link device doesn't. A signal coupler, fieldbus-device power supply, and terminator may be integrated into a single segment coupler.

Bridge

A bridge passes data from one network to the other, thereby allowing devices on different networks to talk to each other (figure 3-30). On one network, all devices have a different node address. Even if they are on different segments within the network there will be no address duplication. However, two devices on different networks may have the same node address; the bridge ensures that the devices are not mixed up. Bridging is very much a software function. It is not really available as a device on its own but rather built into a multi-port linking device or interface card.

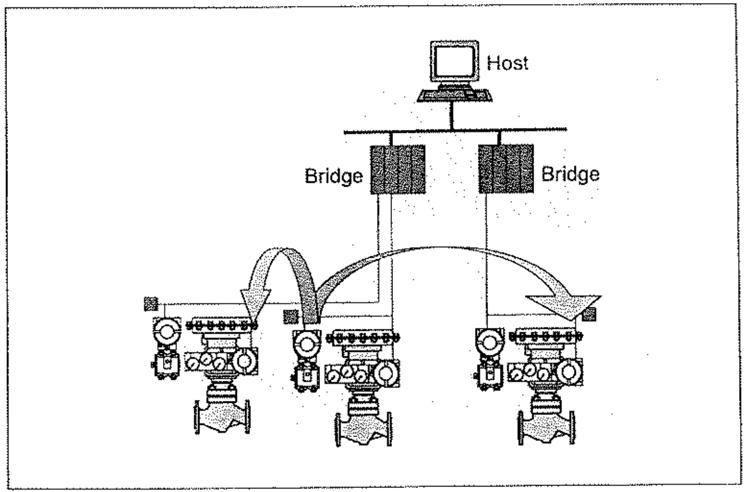


Figure 3-30. A bridge allows devices on different networks to talk to each other.

Regular Power Supply

For networks with bus-powered devices you must connect a power supply to power them. For both FOUNDATION Fieldbus H1 and PROFIBUS PA, the bus-powered devices operate on DC power anywhere between 9 to 32 VDC. Typically, power supplies have an output voltage slightly below 24 VDC. The power consumption is different from one device type to another but is typically is around 12 mA. However, some devices are as high as 24 mA or more. For nonintrinsically safe installations, there is no restriction on the power going into the field. A power supply of 300 mA is more than sufficient for most installations.

The power supply must have output impedance in accordance with IEC 61158-2. A conventional power supply cannot be connected directly to the bus because the output impedance is almost zero, and the communication signal would be short-circuited. Either a purpose-built IEC 61158-2 compliant power supply should be used, or impedance that is compliant with IEC 61158-2 should be inserted between a conventional power supply and the network (figure 3-31). Fieldbus power supplies and impedance modules typically contain the power supply end terminator and are available in configurations for multiple ports such that a single module supports several networks.

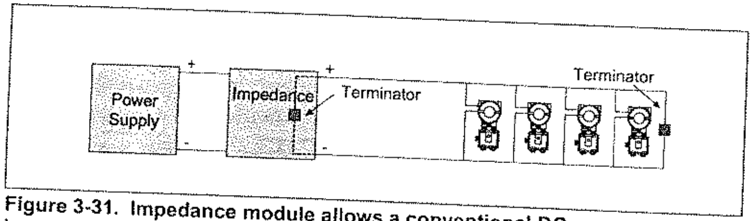


Figure 3-31. Impedance module allows a conventional DC power supply to be used.

Additionally, the power supply and impedance modules normally have a fuse, galvanic isolation, short-circuit protection, and failure indication. For increased availability, the power supply and impedance modules should be redundant and have bumpless switchover. This ensures that the field devices have uninterrupted power even if power supply and impedance module fails. To avoid single points of failure make sure to keep the primary and secondary supplies and impedances separate by some distance. If they are mounted in the same backplane they may be subject to common cause failures. To achieve redundancy, make sure that for the impedance module you use two can be connected in parallel and still maintain compliant impedance. It is important that there be only are two terminators in the network. Built-in terminators are therefore disabled in redundant impedance modules. A separate external terminator is used instead, which also allows either impedance module to be removed for maintenance (figure 3-32).

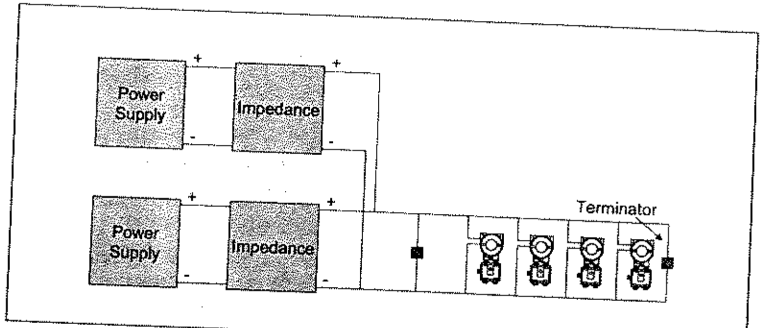


Figure 3-32. Redundant field instrument power.

It's easier to do installation and engineering if the power supply, impedance, and terminator for the field instruments is built into the linking device.

Regular fieldbus power supplies and power supply impedance modules can be used for flameproof (explosion-proof) installations provided that they are mounted outside the hazardous area or inside flameproof enclosures with proper seals and conduits.

WARNING—For flameproof installations, make sure you follow local regulations and codes of practice.

Intrinsic Safety

The concept of intrinsic safety is the same for FOUNDATION Fieldbus and PROFIBUS PA as it is for HART and conventional installation (figure 3-33).

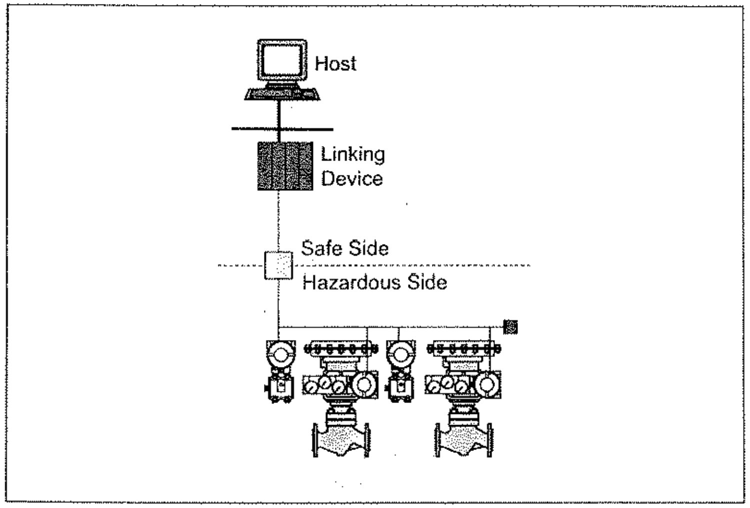


Figure 3-33. The safety barrier separates the safe and hazardous sites.

For IEC 61158-2, all devices operate on 9-32 VDC. Because they are electrically almost identical even for input and output devices, analog as well as discrete, only a single barrier type is required. The main difference between field devices is their power consumption. Because the power available to an intrinsically safe bus is limited, it is important to select devices with low power consumption so as many devices as possible can be connected to each barrier. The power consumption is the main limiting factor to the number of devices that can be used on the intrinsically safe segment. High power consumption can push the limit well below thirty-two devices. However, barriers may be multidropped, which still results in sixteen devices per interface port. The barrier may be a zener barrier or an intrinsically safe galvanic isolator. Intrinsically safe fieldbus devices are current sinking and do not provide power to the network. Even devices that are separately powered draw current from the bus to enable them to communicate.

The number of barriers that are required can be minimized by employing the lowest possible power consumption for the devices.

Terminators contain a capacitor, so they must also be certified intrinsically safe in order to be used in intrinsically safe installations. Since power is scarce, the terminator should be totally passive, with no current consumption, and should not reduce cable length or device count. Since there are different safety barriers for each type of network, it is important to use a safety barrier that has been designed with characteristics as stipulated in IEC 61158-2 rather than some other kind of bus.

WARNING—For the design and installation of intrinsic safety follow local regulations and codes of practice and make sure to read the respective user manuals. Connect only devices certified as intrinsically safe in the hazardous area.

Only one safety barrier can be connected to each hazardous area segment; they cannot be redundant. Typically, the barriers are mounted in the safe area. If they are installed in the hazardous area you need to install a flameproof enclosure with flameproof seals. Take special care to follow the instructions in the product manual for grounding when you are using nonisolated safety barriers. There are two schemes for supplying intrinsically safe power: the traditional entity concept and the newer FISCO model. FISCO provides more power, which means more devices and longer cable can be used.

Entity Concept

The entity parameters for voltage, current, power, capacitance, and inductance are stated in the approval certificate for intrinsically safe devices and barriers. These parameters make it easy to select matching devices and barriers. Since several devices are multidropped off a single barrier you will need to compile the entity parameters of all devices to match against the entity parameters of the barrier. In the traditional entity concept, the cable capacitance and inductance are considered concentrated and therefore must be taken into account when considering the total capacitance and inductance for the hazardous side segment of the network. For the entity concept, barriers with linear output characteristics are used (figure 3-34). For Exia IIC the output power is approximately 1.2W or some 60 mA at 11 VDC. Because of the current limit only a few devices can be connected to each linear barrier. Similarly, the low-voltage output limits the cable length since only a small voltage drop can be accommodated.

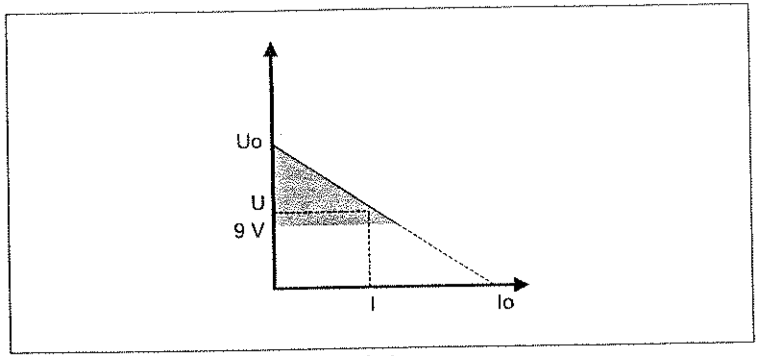


Figure 3-34. Linear output characteristics.

The voltage, power, and current that a field device can handle are limited. Likewise, the total amount of inductance and capacitance allowed on the bus is limited. It is necessary to select a barrier that has voltage, current, and power output that is lower than what is permitted for the field device with the lowest corresponding entity parameter (figure 3-35). The barrier must be able to handle the total external capacitance and inductance of all the devices connected to the safe side plus the network cable. The cable parameters must be obtained from the cable data sheet. Very often, the inductance value for the cable will exceed the value allowed for the barrier. However, a common alternative to the inductance value is to compare the inductance/resistance (L/R) ratio of the cable to the ratio allowed by the barrier. Normally, the cable capacitance is found to be the limiting factor for distance in intrinsically safe installations based on the entity concept. Cables with shield have a higher capacitance than those without. You can calculate the maximum allowed cable distance by referring to the maximum allowed cable capacitance, which is the balance after all the device capacitances have been subtracted from the capacitance allowed by the barrier. An easy way to evaluate the network is to make a table of the entity parameters for all network components (table 3-6).

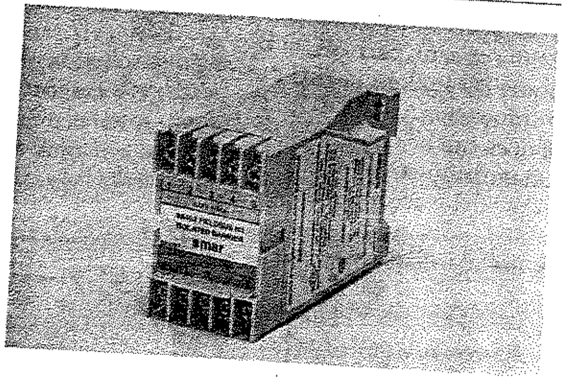


Figure 3-35. IEC 61158-2 safety barrier for entity concept. (Courtesy of Smar)

EXAMPLE 3-6

Typical entity parameters for four devices and cable are shown in table 3-6. In this example, a 500 m cable has typical values of 200 nF/km capacitance and a 25 uH/ohm L/R ratio. The barrier permissible L/R ratio is 30.7 uH/ohm.

Table 3-6. Sample of table for network entity parameter evaluation.

Tag	Ui (Vmax)	Ii (Imax)	Pi (Pmax)	Ci	Li	L/R	Iq
	V	mA	W	nF	uH	uH/ohm	mA
FT-123	24	250	1.5	5	8		12
TT-124	24	250	1.5	5	8		12
FCV-123	24	250	1.5	5	8		12
TCV-124	24	250	1.5	5	8		12
Terminator	24	250	1.5	0	0		0
Wire				100	275	25	0
	min	min	min	sum	sum		sum
	Uo (Voc)	Io (Isc)	Po (Pm)	Co (Ca)	Lo (La)	Lo/Ro (L/R)	
Barrier requirement	24	250	1.5	120	307	25	48
Selected Barrier	21.4	200	1.1	154	300	30.7	60

The output voltage, current, and power of the barrier in this example is lower than the maximum allowed for the field devices. The current provided is sufficient, as is the capacitance allowed. But the inductance allowed is too low. However, the barrier can still be safely used because the L/R ratio is sufficient.

EXAMPLE 3-7

You can calculate the maximum cable length for the barrier in the example due to capacitance to 670 m, by referring to the capacitance left over in the budget after the capacitance of the four devices is subtracted from the barrier limit:

$$L = \frac{C}{c} = \frac{154 - 4 \times 5}{200} = 670$$

Fieldbus Intrinsically Safe Concept (FISCO)

The Fieldbus Intrinsically Safe Concept (FISCO) is a relatively new model for intrinsic safety that was developed by PTB (Physikalisch-Technische Bundesanstalt). In the FISCO model, the cable capacitance and inductance are not considered concentrated nor unprotected as long as the cable parameters are within given limits. For the same reason, FISCO type barriers have no specified permitted capacitance or inductance. FISCO barriers have a trapezoidal output characteristic (see figure 3-36) that provides up to 1.8W of output power for Exia IIC, which in turn makes it possible to connect more devices than is possible for a traditional entity barrier. Most FISCO devices can also be used for entity concept; check the device's various certifications. Not all FISCO approved devices can handle 1.8 W. There are FISCO model barriers available that provide only 1.2 W output power suitable for devices with a low power rating. Be careful to check the power limit of devices. Low power FISCO equipment is often referred to as "small FISCO" or "fisco". Low power fisco barriers power fewer devices but still have the benefit of longer cable and elimination of the need for calculating inductance and capacitance.

FISCO-certified devices have low capacitance and inductance that is considered to be negligible (less than 5 nF and 10 uH, respectively). Cables with parameters within the ranges specified in table 3-7 can be used in FISCO installations for lengths up to 1 km (3,300 ft) with a maximum spur length of 30 m (100 ft). Therefore, there is no need to make the analysis for the cable and devices in the network for these parameters.

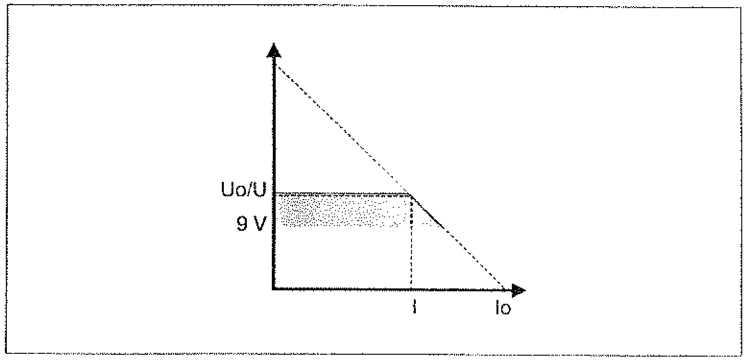


Figure 3-36. Trapezoidal output characteristics.

EXAMPLE 3-8

The number of devices with a typical current consumption (12 mA) that can be connected to a FISCO barrier with a 100 mA output current capacity is eight, which is simply calculated as follows:

$$n = I/i = 100/12 = 8$$

Note that low power 1.2 W fisco barriers have lower output current, typically around 85 mA.

Table 3-7. Permitted FISCO cable parameters.

R (loop)	15-150 ohm/km (4.6-46 ohm/1,000 ft)
L	400-1000 uH/km (120-300 uH/1,000 ft)
C	80-200 nF/km (24-60 nF/1,000 ft)

The effective cable capacitance depends on the connection of the shield and must be calculated using the simple equations in figure 3-37. If the shield is connected to one of the signal conductors at the barrier the capacitance is higher than if the conducts are isolated from the shield connected to earth.

It is necessary to select a barrier that has voltage, current, and power output that is lower than what is permitted for the field device with the lowest corresponding parameter (figure 3-38).

An example of the parameters for eight typical devices is shown in table 3-8.

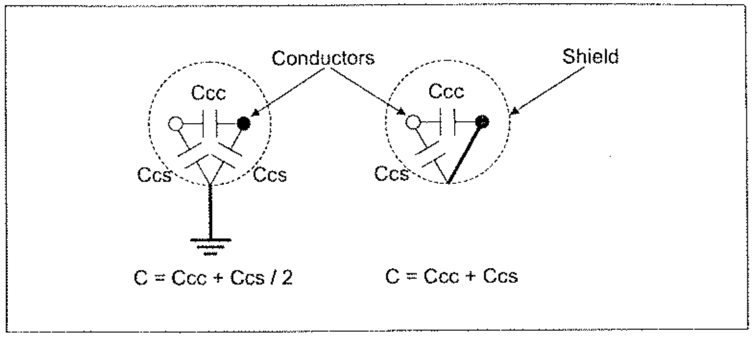


Figure 3-37. Effective cable capacitance depends on shield connection.

Table 3-8. FISCO network analysis.

Tag	Ui (Vmax)	Ii (Imax)	Pi (Pmax)	Iq
	V	mA	W	mA
FT-123	24	250	2	12
TT-124	24	250	2	12
LT-125	24	250	2	12
PT-126	24	250	2	12
FCV-123	24	250	2	12
TCV-124	24	250	2	12
LCV-125	24	250	2	12
PCV-126	24	250	2	12
Terminator	24	250	2	0
	min	min	min	sum
	Uo (Voc)	Io (Isc)	Po (Pm)	
Barrier requirement	24	250	2	96
Selected Barrier	15	190	1.8	100

The output voltage, current, and power of the barrier in this example is lower than the maximum allowed for the field devices. The current provided is sufficient. Thus, the barrier can be used.

The maximum cable length is calculated for voltage drop just as it is for a normal installation because there is no capacitance or inductance limitation up to 1 km as long as the specified cable parameters are followed.

For an intrinsically safe installation with eight devices using a safety barrier with 13.8 V output the maximum distance is calculated as 1.1 km, considering only the voltage drop. However, other aspects set the limit for Exia IIC as 1 km.

Figure 3-38. IEC 61158-2 safety barrier for FISCO. (Courtesy of Smar)

It is important that the barrier and field devices all be FISCO certified. Non-FISCO barriers and devices cannot be used in FISCO-style installations. FISCO field devices must be able to handle the high power output of a FISCO barrier. To be compatible with a typical FISCO barrier the Pi (Pmax) of the device should be larger than the typical 1.8 W provided by the barrier.

Repeating Barriers

Because each safety barrier, and in particular a linear barrier, only connects a few devices it would be rather uneconomical to connect only one barrier per communication port. Some devices have a current consumption that is so high only two devices can be connected to a linear barrier. If you use barriers with built-in repeaters it becomes possible to run one safe area segment from the communication port to several barriers, each one of which has a hazardous area segment. Thus, a larger network is formed and it is possible to have a full sixteen devices per network (figure 3-39). Built-in repeaters eliminate the need for external repeaters.

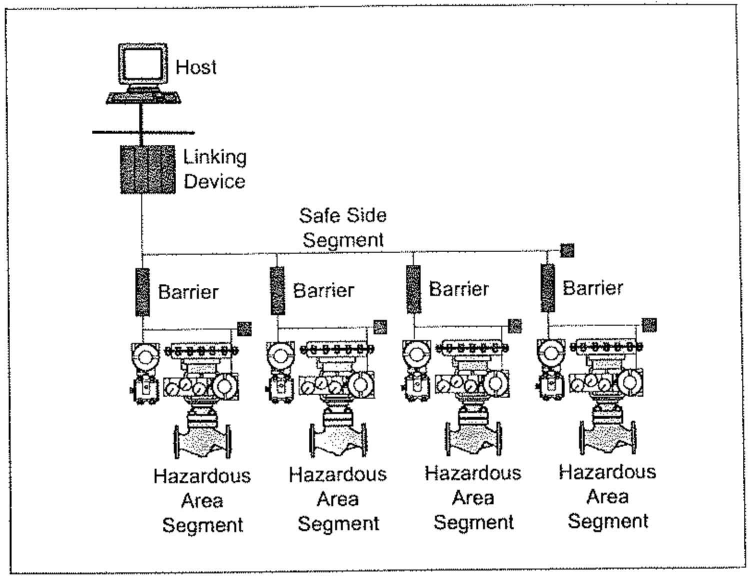


Figure 3-39. Several hazardous area segments connect to a safe side segment using repeating barriers forming a single network.

Intrinsically safe installation becomes simpler and more economical if you use galvanically isolated barriers based on the FISCO model with a built-in repeater and terminator.

Host Preparation

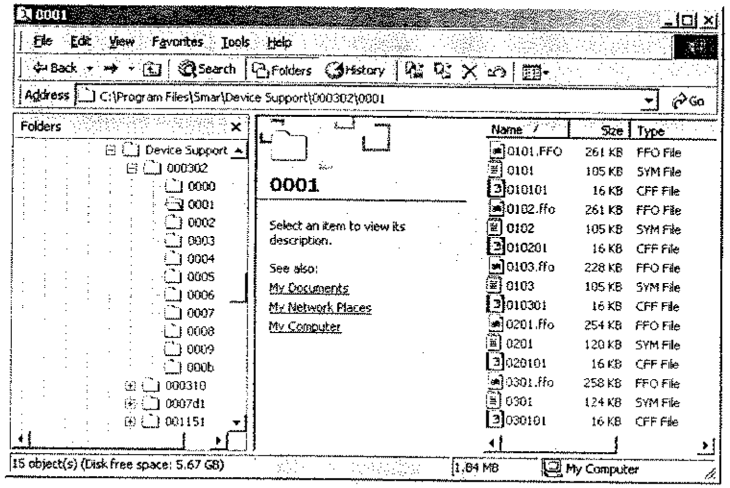
For the host to work with various types of field devices made from different manufacturers the appropriate device support files have to be loaded. If the files are not preloaded in the host, they will have to be installed. Typically, the configuration tool has a dedicated folder for support files with subfolders for each manufacturer and device type (figure 3-40). To prepare the host you should copy the device support files into the appropriate folder. The support files are normally shipped with the devices, or they can often be obtained in advance from the manufacturer's web site. This enables you to start configuration work before the devices are received. As more devices and new revisions become available you must install their support files before they are used. The standard files defined in the specifications that the device manufacturers are required to provide are as follows:

FOUNDATION Fieldbus: Two device description files (FFO and SYM) and one capabilities file (CFF).

PROFIBUS: One device data sheet file (GSD) for class I master and additional files related to class II master. Standard device description files are not yet used.

To enjoy full interoperability it is good idea to choose a host that can support third-party devices without proprietary files from the host manufacturer.

Most configuration tools for FOUNDATION only require the standardized files, but in some systems the host may require additional proprietary files made by the host manufacturer for each device. In the cases where proprietary files are required the supplier would typically have a shortlist of the manufacturers, device types, and device versions that the host supports.



Address C:\Program Files\Smar\Device Support\000302\0001

Name	Size	Type
0101.FFO	261 KB	FFO File
0101	105 KB	SYM File
010101	16 KB	CFF File
0102.ffo	261 KB	FFO File
0102	105 KB	SYM File
010201	16 KB	CFF File
0103.ffo	228 KB	FFO File
0103	105 KB	SYM File
010301	16 KB	CFF File
0201.ffo	254 KB	FFO File
0201	120 KB	SYM File
020101	16 KB	CFF File
0301.ffo	258 KB	FFO File
0301	124 KB	SYM File
030101	16 KB	CFF File

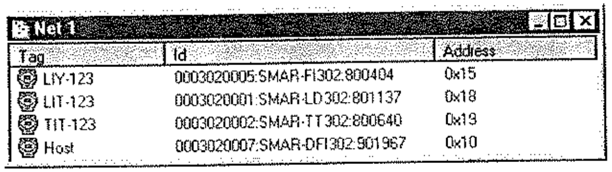
15 object(s) (Disk free space: 5.67 GB) 1.84 MB

Figure 3-40. Three FOUNDATION Fieldbus device support files for each version of each device type.

PROFIBUS is drafting an FDT (field device tool) concept based on Windows software components. An FDT is a frame application in the host from which device configuration will be done. Each device type will have DTM (device type manager) software supplied by the device manufacturer, which should be installed onto the FDT. When a certain device type should be configured the FDT will start executing the associated DTM software. The configuration will then be done from that software.

Device Commissioning

For FOUNDATION Fieldbus the node address assignment is automatically handled by the linking device or other interface so the field device can be connected directly. For PROFIBUS, the device address must be configured manually before the device is installed on the operational network. Once the installed device is connected to the network the linking device or other interface will detect it automatically. The device will then be displayed together with all the other devices on the same network in the live list for each network in the host application (figure 3-41).



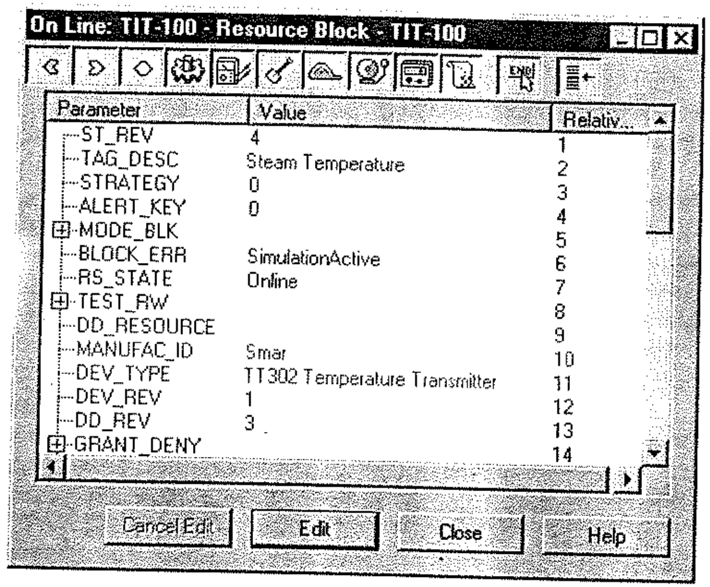
Tag	Id	Address
LIY-123	0003020005:SMAR-FI302:800404	0x15
LIT-123	0003020001:SMAR-LD302:801137	0x18
TIT-123	0003020002:SMAR-TT302:800640	0x19
Host	0003020007:SMAR-DFI302:901967	0x10

Figure 3-41. Live list indicates automatically detected devices.

If the device does not show up in the live list of the engineering tool it is most likely connected to the wrong network. In this case, you can find it in the live list for another network. Another possibility is that the device is not wired or operating properly. You will then need to do some troubleshooting. The live list is thus an excellent tool for commissioning, since it is very easy to confirm proper wiring of the field instruments. To ensure that a device has not been mixed up with another one on the same network that was installed at the wrong location (process point), the device can be temporarily disconnected to confirm that the correct tag disappears from the live list.

It is a good idea to start the device configuration in the engineering tool off line as soon as the detailed engineering is finished, even before the devices arrive on site. Once the devices have been received and connected to the network the ready device configuration can be downloaded immediately.

For further reassurance use the engineering tool to retrieve detailed identification information in any device. This information will enable you to check the tag, descriptor, manufacturer, type, version, and overall health to confirm that it is the correct device and that it is functioning properly (figure 3-42).



Parameter	Value	Relativ...
ST_REV	4	1
TAG_DESC	Steam Temperature	2
STRATEGY	0	3
ALERT_KEY	0	4
⊞ MODE_BLK		5
BLOCK_ERR	SimulationActive	6
RS_STATE	Online	7
⊞ TEST_RW		8
DD_RESOURCE		9
MANUFAC_ID	Smar	10
DEV_TYPE	TT302 Temperature Transmitter	11
DEV_REV	1	12
DD_REV	3	13
⊞ GRANT_DENY		14

Figure 3-42. Detail device identification information found in resource block.

FOUNDATION Fieldbus

The linking device or other interface handles the addressing of all devices on each network completely automatically. For all practical purposes, the node address is hidden and uninteresting to most users, who interact with the system based on tags. However, the node address is typically shown in the live list for use by specialists. As a result, FOUNDATION Fieldbus devices can be commissioned from the control room. Once a device is connected to the network it will be detected within seconds. It is possible to disconnect and connect devices at any time without disturbing the network.

After the device is identified it will automatically be assigned a node address on the network, a process that can take a minute or so. In this process, the device is first given a default node address in the range 248-251 before it receives its final operational address in the range 16-247. The device may disappear from the live list momentarily during this process. The interface detects new devices by continually probing node addresses. To speed up the detection the address range is often narrowed. Thus, if a device is not detected at first you should set the interface to not skip any address so no devices are missed. For FOUNDATION Fieldbus, the device configuration database is prepared based on user-defined device tags without any association with particular physical devices. Therefore, the installed physical devices need to be associated with their respective configuration.

Each FOUNDATION Fieldbus device has a unique thirty-two-character device identifier (Device ID), a kind of hardware address that is totally unique and is used to unambiguously distinguish one device from the others. Like HART and Ethernet, this address is set in the circuit board by the manufacturer and cannot be changed. A field device is associated to its particular configuration by correlating the device configuration tag with its device ID. This is a simple point-and-click operation since all the IDs of the devices on the network are already in the host live list. If the device tag in the instrument already matches a tag in the configuration the two tags will automatically be associated. Once attached to the network the physical device tag can be assigned, if it hasn't already.

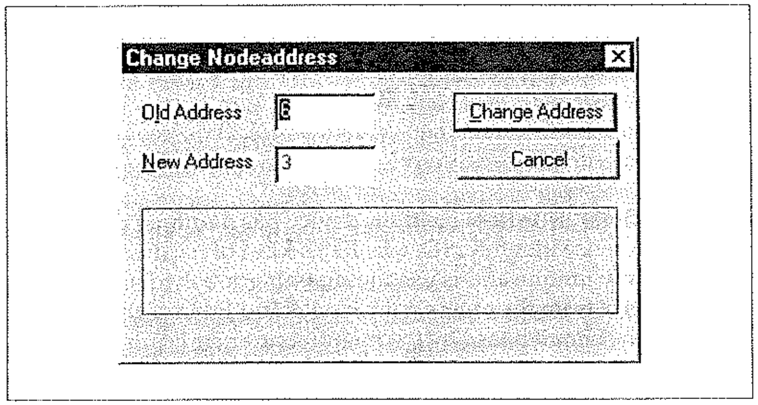
Save time by ordering instruments that have the device tag preconfigured by the factory so they will automatically be matched to the configuration.

Once all the devices are associated with their respective device tags the configuration can be downloaded, either for the entire network at once or individually.

PROFIBUS PA

If the address is not set correctly the device will not be detected and may even disrupt the communication of the PA devices already on the network. The device address can either be set locally at the device or remotely using the configuration tool. Depending on the device, local setting may either be done in the device using internal hardware DIP switches or, in the case of some devices, externally from the local digital display using local adjustment keys. Remote setting is done one device at the time using a configuration tool that is connected to a single device on a separate network so as to avoid disrupting the operational networks (figure 3-43).

The address must be unique on the network. It is therefore important to take measures to prevent any address duplication. Duplicate addresses may cause control to be disrupted. It may therefore be a good idea to have a tag and address cross-reference list for



Old Address: 6

New Address: 3

Figure 3-43. PROFIBUS-PA device address setting.

each network as a part of the system documentation. Valid addresses are in the range 0-126.

Ethernet and IP (FOUNDATION HSE and PROFInet)

In most cases, Ethernet wiring and hardware access go hand in hand with the Internet Protocol (IP) for networking and the Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) for data transport between nodes. Ethernet is by far the most common form of wiring for LANs, but it is not always used with TCP/IP. Likewise, TCP/IP is not always used with Ethernet because it travels just as well on RS232 or on the public telephone network. Both FOUNDATION Fieldbus HSE and PROFInet—along with many other protocols, including Modbus/TCP—have wiring and hardware access that is based on Ethernet IEEE 802.3 / ISO 8022.

Since they are built on the same platform Ethernet devices with different protocols can coexist on the same network without conflict (though they do not talk to each other because of differences in the user layer). The electrical installation is therefore identical. It should be noted that both Ethernet and IP standards have addressing schemes, called Ethernet MAC address (which is a hardware address) and Internet IP network addresses, respectively. The device manufacturer makes the Ethernet MAC address unique and sets it in the hardware. As a result, there are no two identical MAC addresses. The unique hardware address consists of a centrally administered manufacturer code and a unique identifier. This ensures that no two addresses in the world are the same, which eliminates the chance of conflict. The user can set the IP network address or it can be assigned automatically.

Ethernet Terminology and Basics

In the past, Ethernet had a multidrop bus topology using coaxial cable media. For several years now, however, the twisted-pair cable media in a star topology has been the most common wiring scheme. Though Ethernet is constantly getting faster, for process control devices it primarily comes in two speeds, 10 Mbit/s and 100 Mbit/s, with the latter called "Fast Ethernet." The twisted-pair version is referred to as 10Base-T for the 10 Mbit/s version and 100Base-TX for the 100 Mbit/s version. The fiber-optic media is referred to as 10Base-FL for the 10 Mbit/s version and 100Base-FX for the 100 Mbit/s version. Each twisted-pair segment has only two devices connected to it, one in each end. A simple network may consist of only a single segment connecting, for example, a workstation and a linking device (figure 3-44).

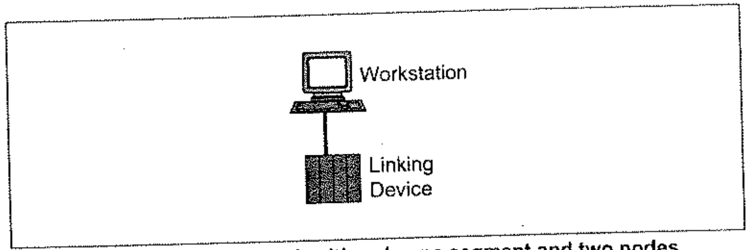


Figure 3-44. A simple network with only one segment and two nodes.

In most installations many segments are used. Using a star topology, multiple segments are joined at a hub to form a larger network (figure 3-45).

Devices on the Ethernet are powered separately and are not intrinsically safe. It is possible to connect or remove a device while the bus is running.

Ethernet Topologies and Components

Star topology is now the most commonly used formation for Ethernet. Every node is connected to its own port on the central hub and has its own exclusive segment running like a spoke from the hub (figure 3-46). A major advantage of this formation is that a failed node or wire segment does not affect the other nodes, which results in high availability. Devices can be installed and removed

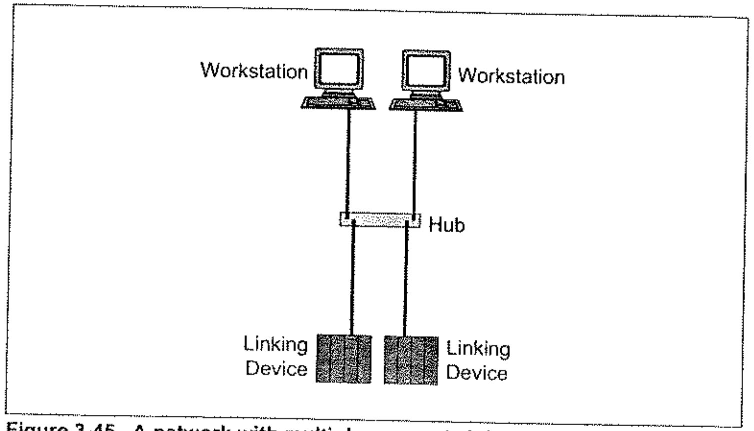


Figure 3-45. A network with multiple segments joined at a hub.

under power without disrupting other nodes. Because of the high speed, the twisted-pair cable is limited to 100 m (330 ft).

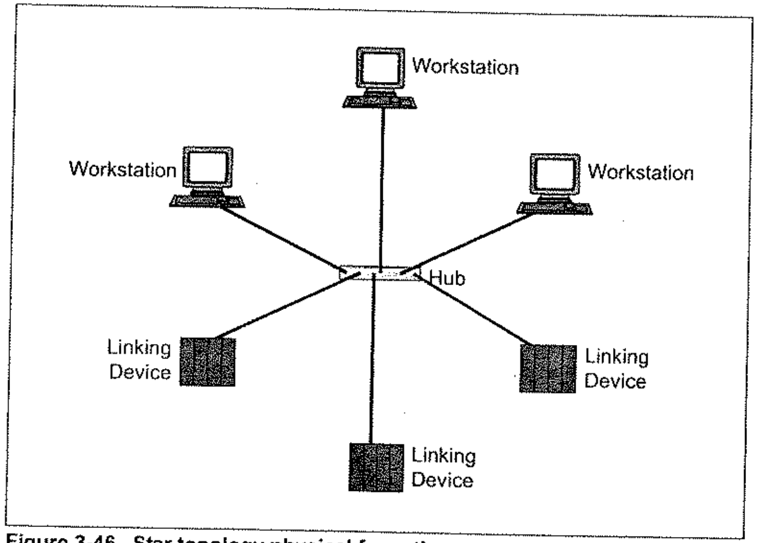


Figure 3-46. Star topology physical formation.

Although the network formation is a physical star, logically it is a bus structure. It is therefore often drawn as a traditional multidrop bus with the hub omitted to make the architecture easier to visualize (figure 3-47).

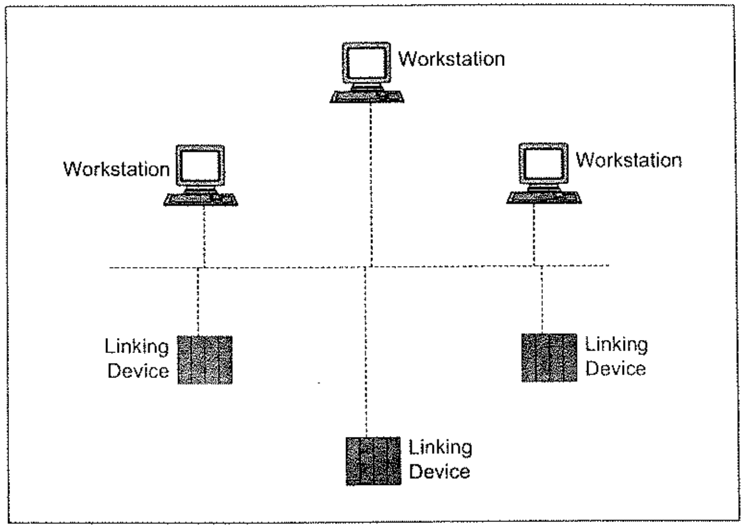


Figure 3-47. Star formation logically represented as a bus topology.

Because of the signal attenuation along the wires there is a limit to how long a network segment can be. There are two types of hubs, which have significant differences: shared hubs, often called just "hubs," and switched hubs, often called just "switches." Shared hubs and repeaters connect segments, whereas switched hubs and bridges connect networks. An LED on the hub indicates valid connection.

Shared Hub and Repeater

A shared hub is essentially a multi-port repeater that refreshes the timing of the signal received on one port and retransmits the refreshed signal on all its other ports. The hub is bidirectional. Hubs may thus be used to connect segments together to form a larger network. When the nodes of hubs are connected in a star formation, hubs may be cascaded to form a tree topology, which allows the network to be expanded to accommodate more nodes (figure 3-48). Because additional hubs can be connected under power the Ethernet architecture is extremely scalable.

For 10 Mbit/s there may be no more than four repeaters or shared hubs between any two devices on the network. That is, a maximum of five segments separate two devices (figure 3-49). Hubs can also be connected in series to overcome the attenuation associated with long distances. If four repeaters are connected in series a total

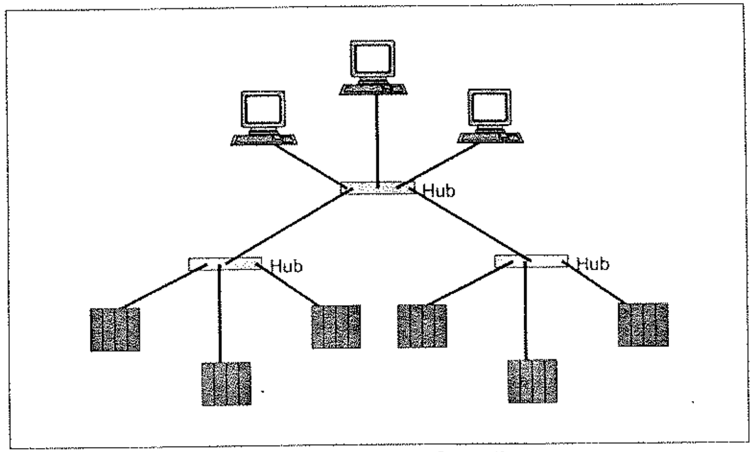


Figure 3-48. Tree topology from joined star formations.

distance of five times 100 m (330 ft), a network diameter of 500 m (1600 ft) can be achieved.

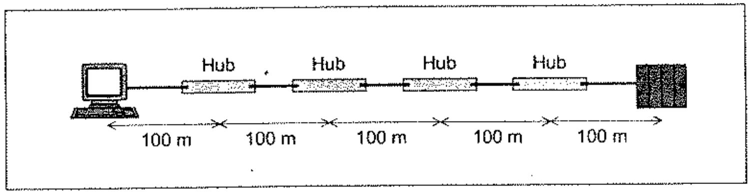


Figure 3-49. For 10Base-T the maximum network diameter is 500 m.

When you use shared hubs in a tree formation to increase the number of nodes on the network don't forget that there may be a maximum of five wire segments separating any two nodes. This means a maximum of four shared hubs can be connected together between any two nodes (figure 3-50). For 100 Mbit/s repeaters are classified according to their latency. Class I repeaters are slower, and the network is therefore limited to only one per network. Class II repeaters are faster, but there can still only be two of them, and the network diameter is limited to 205 m (673 ft).

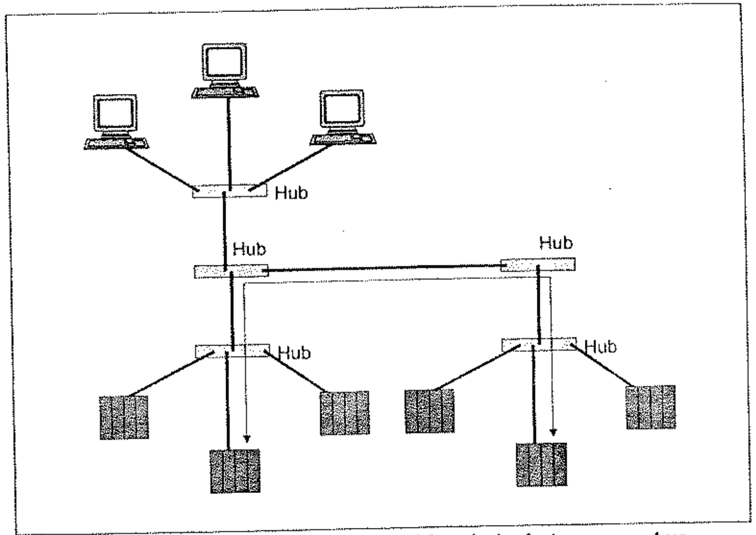


Figure 3-50. There shall be a maximum of four hubs between any two devices on a 10 Mbit/s network.

The shared hub broadcasts a message received on one port on all the other ports. When shared hubs are used the network bandwidth is shared among the users, and each node gets only a part of the available bandwidth. When one node is using the network another node cannot access it. However, for shared Ethernet there is a small risk that two nodes will attempt to use the network at the same time, causing a communication collision. In this case, they will retry again a fraction of a second later, but not at the same time. All the nodes that share the same network are said to be in the same collision domain. Because of the possibility that a random delay may be introduced a shared Ethernet is said to be nondeterministic. The possibility of a collision increases with the number of nodes and the network traffic. Collisions therefore put a practical limit on the number nodes that can share a network, that is, the number of nodes in the same collision domain.

Hubs have the intelligence to monitor the communications on each port. If too many collisions occur the port is cut off or partitioned. The hub rejects messages to prevent the maximum allowed by the standard from being exceeded and to prevent jabber.

Switching Hub and Bridge

An Ethernet bridge connects smaller separate Ethernet networks together to a larger Ethernet network (figure 3-51). These could be networks operating at the same or different speeds and on the same or different media. The bridge has "intelligence" for keeping track of the Ethernet MAC address of the devices on the local network. It typically learns these automatically by watching the traffic and therefore does not require any configuration. The bridge acts as a filter in which only those messages with Ethernet addresses destined for nodes not on the local network are forwarded to the remote network. A bridge receives the entire message and checks it before retransmitting it. Erroneous messages are rejected. Bridges operate independently of the protocol used and are therefore very versatile.

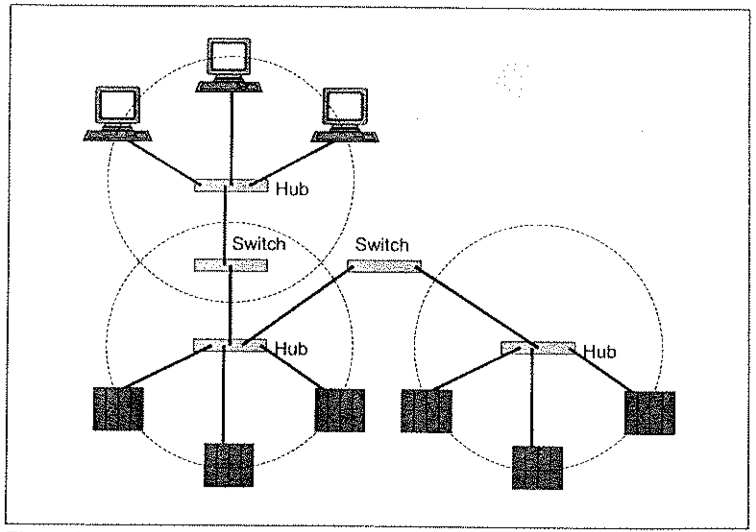


Figure 3-51. A switched hub connects separate networks together.

A switching hub is different from a shared hub in that it does not broadcast received messages on all its ports. A switched hub has the intelligence to examine the Ethernet destination address for the message and only transmit the message on that one corresponding port. Most switches are self-learning, that is, they automatically establish what addresses are on which port. This characteristic offers many benefits. The switching hub supports multiple simultaneous bridging operations that connect several networks together. As a result, communication between other ports can occur concurrently without each interfering with the other (figure 3-52).

If a device on a network connected to port 1 talks to a device on a network on port 5 the messages are filtered and only exchanged between these two ports. Thus, they don't collide with messages exchanged between ports 2 and 3.

The intelligent internal cross-connection increases the available bandwidth since there are several simultaneous communications.

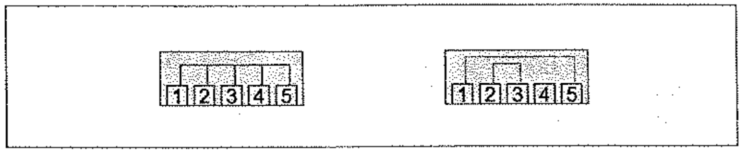


Figure 3-52. Shared hub (left) vs. switched hub (right).

Moreover, the network is divided into separate smaller collision domains. With fewer nodes, the chances of collisions decrease or are essentially eliminated, making Ethernet deterministic. Networks based on switched hubs can therefore grow much larger than the practical limitations imposed by shared hubs.

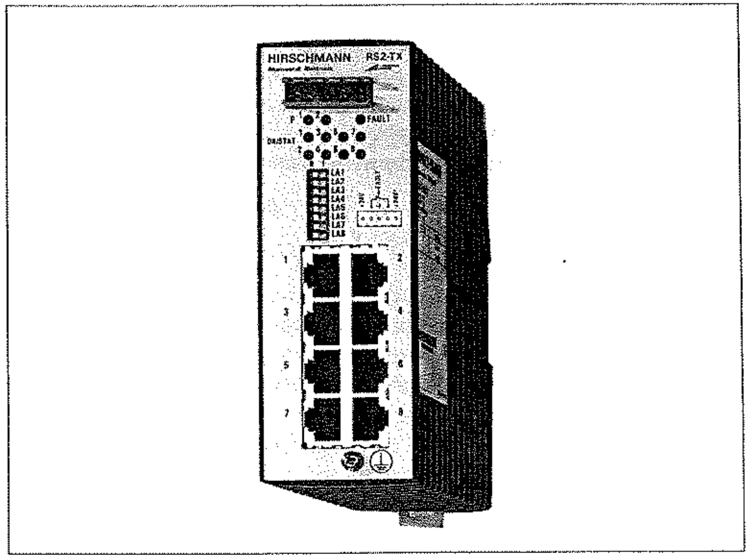


Figure 3-53. Industrial grade 100Base-TX switching Ethernet hub with RJ45 connectors. (Courtesy of Hirschmann)

Shared and switched hubs for 10 Mbit/s and 100 Mbit/s are available in both low-cost commercial-grade versions and in industrial-grade versions. In some cases, these hubs can be assigned an IP address, and network management information such as communication statistics can be retrieved. Industrial-grade hubs have extended temperature range, are DIN-rail mounted, have failure indication contact output for safety, and may have redundant power supply for higher availability (figure 3-53).

Use switching hubs for best performance and choose the industrial-grade version for reliability.

Some Ethernet devices in the plant may be communicating at 10 Mbit/s and others at 100 Mbit/s. Some switching hubs have an auto-negotiating function. This allowed them to automatically detect the speed on the network connected to each of its ports and adjust itself accordingly. Messages are stored internally before forwarding, which solves the problem of speed differences between networks.

There is no limit to the number of bridges and switches that can be cascaded as there is in the case of shared hubs and repeaters. The reason for this is that other devices communicate with bridges and switches, not through them, as in the case of shared hubs and repeaters. Therefore, bridges and switches are not delaying the Ethernet signal.

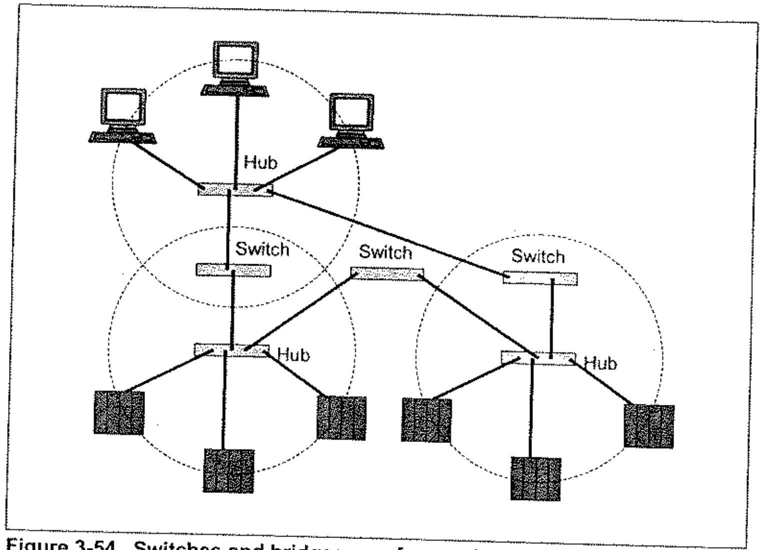


Figure 3-54. Switches and bridges may form redundant circular paths between networks.

Ethernet can be made deterministic by using 100Base-TX and switched hubs and by maintaining a maximum 50 percent loading through the use of few nodes per network. In this way, the chance of a message being delayed is less than the chance of a message being lost due to noise.

When several switches or bridges interconnect networks there is a chance that redundant paths link some of the networks, which then form a loop (figure 3-54). To avoid duplicated messages being forwarded by switches in opposite directions in the loop, a switch must support the spanning tree algorithm. This algorithm prevents a loop formation by disabling one of the bridges. Should a failure occur in a switch the disabled bridge is automatically enabled after some time is spent restoring communication.

Switching hubs learn which port is a particular node's destination Ethernet address by flooding the first message for an address on all ports and then waiting to see which port a reply is received on. This MAC address learning method works fine when not too many nodes are involved. Bridges and switching hubs are therefore suitable for linking smaller networks, but when larger networks are joined it is better to use a router. Similarly, bridges and switches forward broadcast messages throughout the networks, which will adversely affect performance, depending on the amount of broadcast traffic. It is recommended that you partition networks using routers for broadcast containment when you expect a lot of broadcast traffic.

Network Interface Card

Computer workstations, servers, and portables are connected to the Ethernet through a network interface card (NIC). NICs come in several different styles, including internal ones that plug into a slot on the motherboard or into the side of portables in the form of a PCMCIA (PC card). NICs normally support both 10 Mbit/s and 100 Mbit/s and automatically select the appropriate speed.

Transceiver

A transceiver is media converter that is used to convert, for example, from twisted-pair cable into fiber optics (figure 3-55). Transceivers are also available to convert older forms of Ethernet like 10Base-2 into 10Base-T.

Ethernet Media

Twisted-pair media use two pairs, one for transmitting and the other for receiving. The two pairs are running inside one and the same cable, which makes connection easy. Standard Ethernet twisted-pair cable contains four pairs but only two are used. Similarly, the fiber-optic media use two strands, one for transmitting and the other for receiving. Standard Ethernet optic cable contains both strands in one cable. When two nodes are connected directly

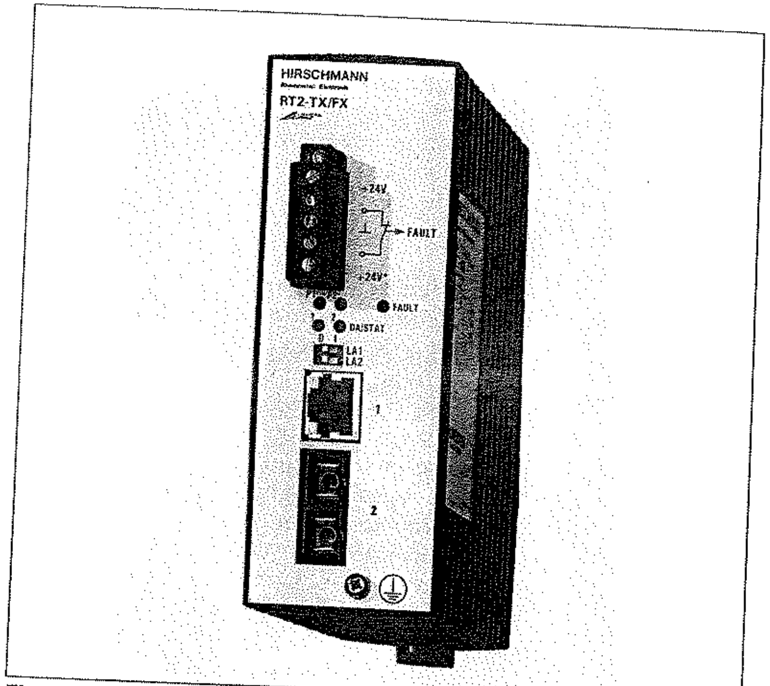


Figure 3-55. Industrial Ethernet 100Base-TX to 100Base-FX transceiver. (Courtesy of Hirschmann)

the transmit from one node shall be connected to the receive on the other node. This is called a crossover connection, and special crossover cables are available for a direct connection between nodes. However, a hub takes care of the cross connection internally, so the nodes in networks based on hubs connect to the hubs through a straight-through cable. When two hubs are connected together a crossover cable is also required between them, unless the hub has a dedicated uplink port, which has this crossover built in (figure 3-56).

To make the networking easier to administer it is a good idea to use a patch panel at the hub to facilitate changes. Use patch cords to establish desired connections. For large systems hubs can be installed in a dedicated closet. In the control room, workstations can be connected through a telecommunications outlet using work area cables (figure 3-57). A similar scheme may be implemented inside the panel to make it easy to connect and disconnect the Ethernet devices.

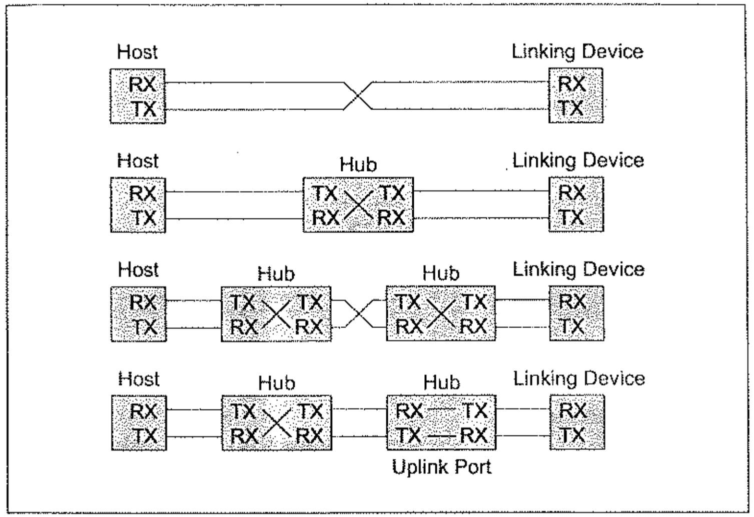


Figure 3-56. Connection of node-node, node-hub, hub-hub and hub-hub with up-link port.

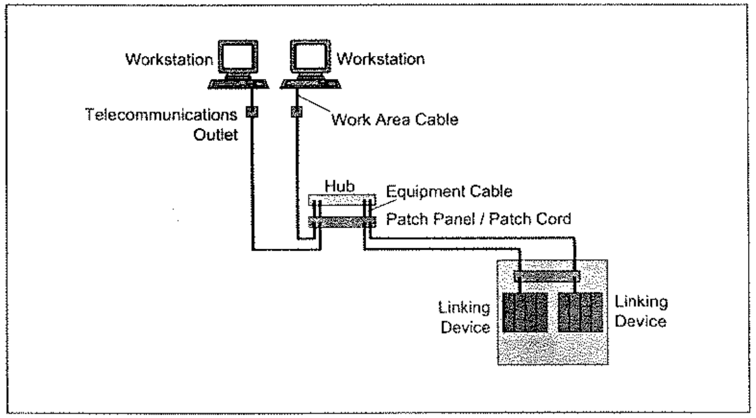


Figure 3-57. Ethernet cabling model.

Out of the 100 m (330 ft) of cable that is allowed for the segments a maximum total length of 10 m (33 ft) can be used for patch cords and work-area cables. Good cabling practice should be applied, which includes identifying cables through labeling and proper documentation.

Copper Wire

There are two types of twisted-pair copper wire cables for Ethernet: UTP, which has four unshielded twisted pairs, sometimes with an overall shield, and STP, which has four individually shielded twisted pairs with an additional overall shield. For 10Base-T and 100Base-TX only two of the pairs are used. Two common categories of cabling are used for Ethernet in the EIA/TIA-568 standard. UTP category 3 is suitable for 10 Mbit/s, and category 5 is suitable for 100 Mbit/s or lower. For twisted-pair, no couplers like the ones used in coax cable are required. Devices and cables have RJ45 plug-in connectors. Only a minimum of cable jacket should be removed for termination to the connector, and twisting should be maintained within 13 mm (a half inch) of the ends to eliminate noise pickup. Placing UTP cable in a metal conduit may result in degraded performance. If the cable is to be run in harsh environments use special cable with higher pull strength and a heavier chemically resistant jacket (figure 3-58).

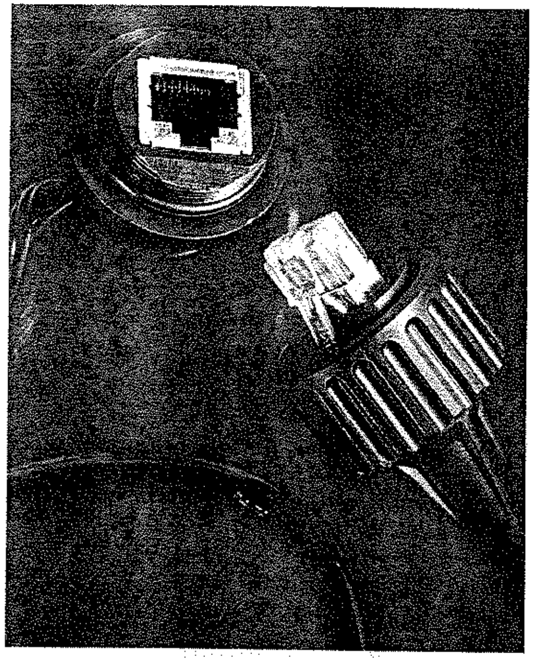


Figure 3-58. Molded industrially hardened RJ45 connectors. (Courtesy of RJLNxxx)

In an industrial environment it may be a good idea to use rugged connectors capable of withstanding vibration, shock, water, and dust.

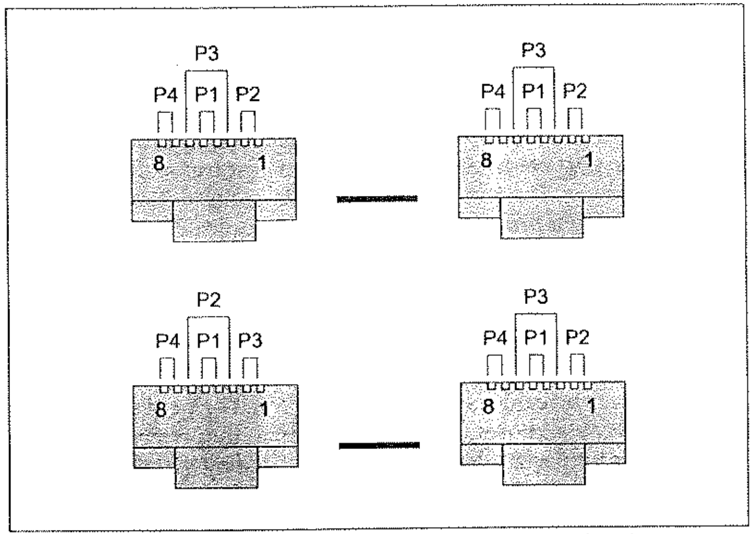


Figure 3-59. Wire pair arrangement in straight through (top) and crossover (bottom) cable (front view).

Table 3-9. Pin assignments for straight through and crossover connectors.

Pair	Pin	Color	Assignment
1	5	White/Blue	Unused
	4	Blue	Unused
2	3	White/Orange	Rx+
	6	Orange	Rx-
3	1	White/Green	Tx+
	2	Green	Tx-
4	7	White/Brown	Unused
	8	Brown	Unused
Crossover (A)			

Pair	Pin	Color	Assignment
1	5	White/Blue	Unused
	4	Blue	Unused
2	1	White/Orange	Tx+
	2	Orange	Tx-
3	3	White/Green	Rx+
	6	Green	Rx-
4	7	White/Brown	Unused
	8	Brown	Unused
Straight through (B)			

The Ethernet signal is polarized, but most devices have circuitry that detects and corrects signal polarity. The preferred pin assignment for new installations is shown in table 3-9 and figure 3-59, but variations exist. It is very important that the pairs are not split, as this would disable the noise cancellation effect, resulting in poor or no communication. Therefore, pairs cannot be, for example, pins 3-4 or 5-6. Ready-made cables with molded connectors save time since connector crimping is not required, but they may be a bit more difficult to run through conduits.

Optic Fiber

Optical fibers are not electrical conductors and are therefore immune to electrical noise and resistant to problems caused by lightning and differences in ground potential. They are therefore ideal for long distances, for example, between buildings. A fiber-optic port connects two strands, one for transmitting and one for receiving. The two strands are often included in the same cable, one with a protective sheath, called duplex cable.

There are two types of optical fibers: multi-mode and single-mode. The fiber core is made from glass and is very thin, typically 62.5 µm in diameter, and surrounded by a cladding of 125 µm for multi-mode. This is usually expressed as 62.5/125 µm. Single-mode fiber-optic cable is usually 10/125 µm. For new installations, it is preferable to use duplex connectors with physical keying for consistent polarization. The maximum distance depends on transmitter power and receiver sensitivity and several factors in between such as cable attenuation ("loss"), connectors, and splices. Multi-mode fiber normally has an attenuation of 3.5 dB/km, typically reaching some 2 km. Single-mode fiber has much lower attenuation, normally as low as 0.3 dB/km, which allows segments as long as 40 km. Single-mode fiber is cheaper than multi-mode fiber, but the devices are more expensive. Most devices do not have a fiber-optic connection, so a transceiver has to be used (figure 3-60).

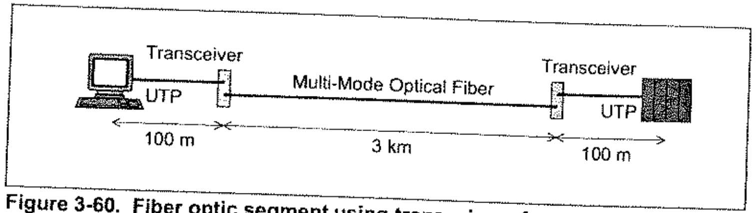


Figure 3-60. Fiber optic segment using transceivers for longer distance.

The ratio of transmitter power and receiver sensitivity is called link budget. The link budget is used to analyze the segment to ensure that there is enough power to exceed the losses along the cable. It is good practice to also leave a margin of 3 to 6 dB to handle performance degradation caused by aging and repairs. Intermediate connectors also create a loss.

For example, the 3 km optical segment in figure 3-60 uses multi-mode fiber with an attenuation of 3.5 dB/km. Using transceivers with a 14 dB link budget, the margin becomes 3.5 dB:

$$14-3\times3.5=3.5dB$$

If single-mode fiber is used instead, with 11 dB link budget transceivers and maintaining a margin of 3.5 dB, the maximum distance becomes 25 km:

$$\frac{11-3.5}{0.3}=25km$$

Care should be taken with short optical cable because the receiver may be overloaded. Fiber optics is often used to link shared or switching hubs together.

Fiber optics creates no spark and can therefore be run through hazardous areas. However, care must be taken since a too-powerful light emanating from a broken fiber could illuminate dust and heating it to the point of ignition. For single-mode, a laser light source is often used. Manufacturers have developed a number of solutions for automatically turning off the transmitter if the cable breaks.

Whenever there is a risk of electrical transients, make sure to remove any metallic shield in the fiber-optic cable within a distance of one meter from the equipment.

Wireless

For long distances, wireless Ethernet using radio is an alternative to fiber-optic cable and leased lines. Wireless is ideal where phone lines are not available, and it avoids obstacles such as crossing land owned by others, roads, rivers, and the like. Wireless is often more economical since it can be very costly to do the trenching to bury cable underground, and leased lines may have high monthly subscription fees. Thus, radio is an ideal option for remote applications.

An Ethernet radio behaves like a bridge linking two or more Ethernet networks together (figure 3-61). In one end of the radio there is a RJ-45 connector for the UTP cable and in the other a connector for the antenna. The radio is totally transparent, and other Ethernet devices do not perceive it since it appears like any other bridge.

The standard for wireless Ethernet is the IEEE 802.11, but proprietary solutions also exist. Though many IEEE 802.11 bridges are available, most are for indoor use and unsuitable for outdoor applications. Rugged outdoor versions generally employ the frequency-hopping option in the 2.4 GHz band, which requires no user license. IEEE 802.11 supports 1, 2, and 11 Mbit/s bit rates. The range depends on many factors, including radio transmission power, receiver sensitivity, and antenna gain. The typical range is 3-6 km, but conditions and equipment may allow for more. A free line of sight is also required.

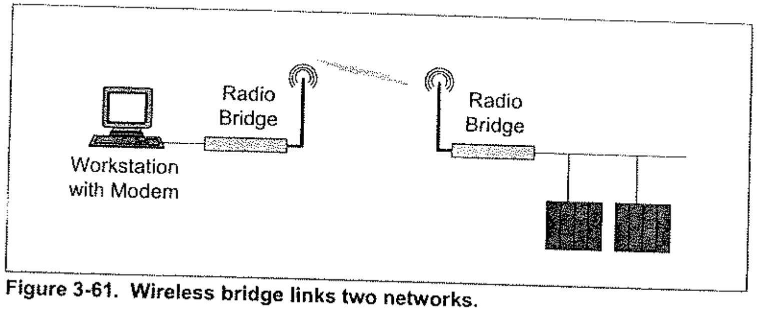


Figure 3-61. Wireless bridge links two networks.

This book does not cover the basics of radio telemetry such as antenna selection and installation. The general rules for radios apply. Refer to *Supervisory Control and Data Acquisition* by Stuart A. Boyer (2d ed., ISBN 1-55617-660-0) for more information.

Ethernet Surge Protection

For most applications, the Ethernet runs within one and the same building, with no direct connection to the outside world. It is therefore not being subjected to the dangers of great earth potential differences. When communication between buildings is required it may be a good idea to use fiber-optic cable instead of copper wire. However, if copper wire is used between buildings or locations where ground potential difference or other sources of surge may occur, you should use surge-protection devices. Good surge-protection devices survive repeated surges and let the protected devices resume normal operation once the surge is over. Install the Ethernet surge-protection device in both locations, and let the network enter the building close to where the main electrical supply earth enters (figure 3-62).

LAN surge-protection devices are available in a single-port configuration that is suitable for remote-mounted linking devices

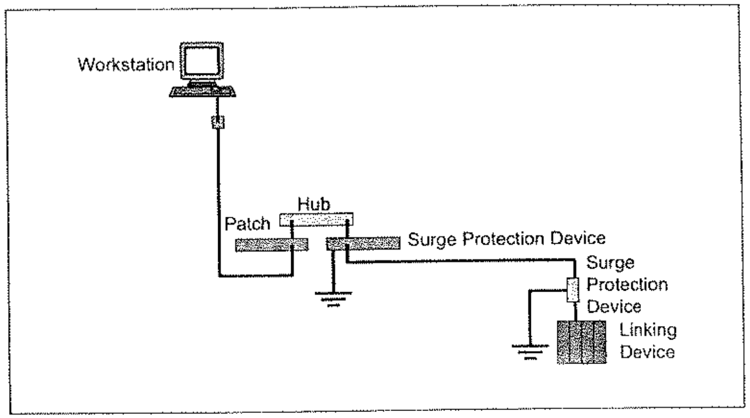


Figure 3-62. LAN surge protection.

(figure 3-63). They are also available in multi-port configurations that are suitable for protection at the host end where the network enters the hub. They take the place of the regular patch panel (figure 3-64).

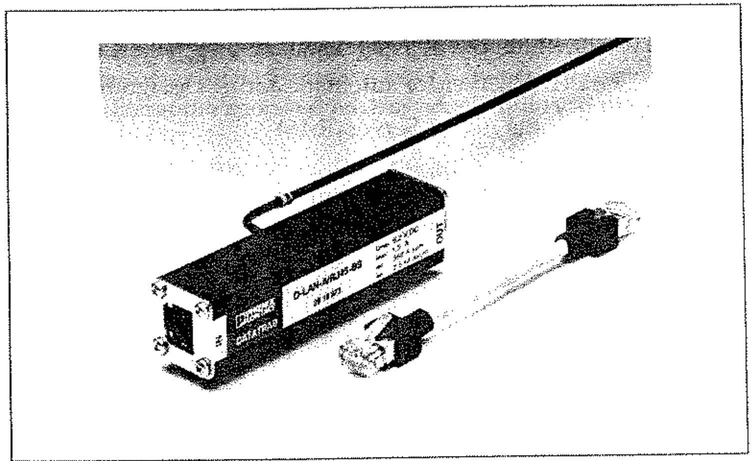


Figure 3-63. Ethernet surge protector. (Courtesy of Phoenix Contact)

IP Terminology and Basics

Both FOUNDATION Fieldbus HSE and PROFInet are based on the Internet Protocol (IP) RFC standards 791. HSE and PROFInet devices can exist on the same network but not talk to each other directly. FOUNDATION Fieldbus HSE devices use the UDP transport

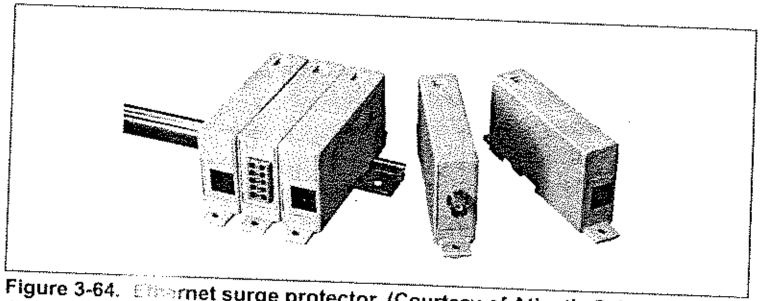


Figure 3-64. Ethernet surge protector. (Courtesy of Atlantic Scientific Corporation)

protocol, whereas PROFInet and Modbus/TCP use the TCP transport protocol. The IP address in each device can be configured and is usually assigned by the corporate network administrator.

On the host-level network the main devices are linking devices, workstations, and servers. In IP terminology, all nodes on the network are called hosts. Servers are software applications, and several typically run on one and the same computer. In an office environment the most common types of servers are file servers and printer servers. Some of the several types of server applications typically found in a process control system are the following:

- DHCP server for automatic address assignment
- Remote Access Server (RAS)
- OPC server for data access

The Internet is one huge network formed by the interconnection of several networks around the globe. A portion of the IP network address denotes which of the many networks the node is on, and the remainder of the address is denoting the host (network node) itself. Think of each one of all the different small networks that are connected to form the Internet as having a network identification number. Each computer or other node on each network has a host identification number. These are the two numbers that make up the IP address. For IP version 4, the address is four bytes, which are typically presented as separated by dots, for example, 209.196.30.107. To avoid conflicts on the Internet the IP address network numbers are centrally administered. However, IANA (Internet Assigned Number Authority) has allocated a few network numbers for free use, which has resulted in a few blocks of private IP addresses that can be assigned to devices without applying. These private addresses are used when a direct permanent connection to the Internet is not required and are always filtered out from the Internet to avoid conflict.

The process control system host-level network is typically not connected directly or permanently to the Internet; that is, it is a private network. Therefore, one of three free IP network numbers is generally used for the host-level network, and an IP address from the associated blocks of address space is assigned to each device on the private network (see table 3-10).

Table 3-10. Private IP address space blocks.

Start Address	End Address
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

IP Router

A router is also a device that connects networks together. It therefore performs a function somewhat similar to a switching hub or bridge in the sense that it filters communications. The automatic MAC address-learning method used by switching hubs and bridges becomes inefficient when large networks are connected together. For applications when a large network is connected to another, a router should be used. A router is network protocol-dependent because the filtering is based on the network address. Therefore, an IP router is required for IP networks because it makes its forwarding decisions based on the IP network address rather than the Ethernet MAC address used by switches and bridges. Because both HSE and PROFInet use IP they both use IP routers.

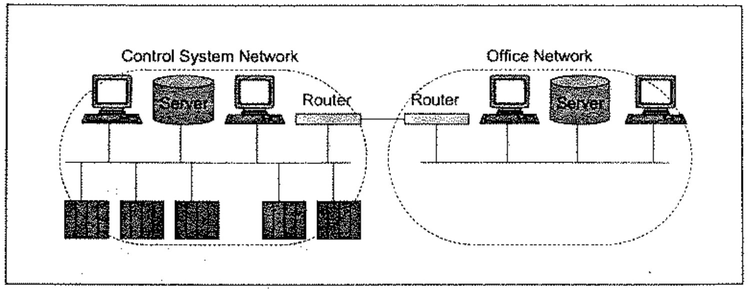


Figure 3-65. Routers physically link network while at the same time creating logically separated subnets.

For example, host-level Ethernet in the control system and the office Ethernet may be physically connected to allow some data to be exchanged while at the same time ensuring that office network traffic does not load the control system network and vice versa (figure 3-65).

When a network becomes large it can be partitioned using routers to improve performance by localizing much of the traffic on smaller networks. A router has an Ethernet MAC address, and when a node on one network wants to send a message to a node on another network the message is first sent to the router. When the router receives the message it examines the network addresses and forwards the message to the router on the appropriate network. The router on the destination network passes the message to the final node. Routers are typically configured manually at the time of installation. A router is used to permanently connect a network to the Internet.

A "brouter" is a hybrid of a bridge and a router. If the router function supports the network protocol of a frame that it treats it acts as a router would; otherwise, the frame is treated as by a bridge.

Gateway

A gateway performs complete protocol conversion and can therefore be used to link together networks with different protocols. A gateway is protocol specific and therefore must be purchased for the desired combination of protocols, for example, Modbus or some other protocol to FOUNDATION Fieldbus HSE. A gateway function may be built into another device as an auxiliary function, for example, into a linking device. Routers are often erroneously called gateways. It is therefore important to be clear about terminology when discussing gateways.

Typically, a gateway has to be manually configured by mapping the device address, memory register, data type, scaling, and other information in the foreign protocol to the desired representation for every parameter that has to be communicated. For example, for the FOUNDATION Fieldbus protocol information should be represented as function blocks. This can be a rather tedious process that requires careful test and verification to make sure it has been done correctly. As far as possible, a single protocol should be used throughout the system to avoid the need for gateways. However, existing plants always have legacy devices that need to be integrated, and even for new plants some equipment may only be available with foreign protocols. For example, Modbus is a well-entrenched protocol in the process industries, and it is important for any system to have a gateway to Modbus.

OPC Server

Most software applications, with the exception of configuration and maintenance tools, do not support protocols like FOUNDATION Fieldbus HSE or PROFInet directly. OPC (OLE for Process Control) is a technology that enables hardware and software to exchange data. OPC is explained in chapter 7, "Operation." Most modern devices come with an OPC server for each type of device or protocol. Similarly, most software applications are OPC clients. The OPC servers are device- or protocol-tailored software programs that provide the device data in a standard format for the device- and protocol-independent OPC clients that are making requests to read and write. This is called a client/server relationship (figure 3-66). For example, a linking device comes with an OPC server, and a process visualization software is an OPC client.

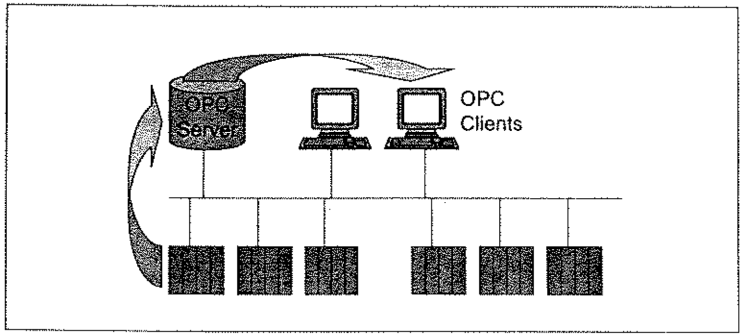


Figure 3-66. OPC client-server relationship.

There are OPC client/server relationships tailored for simple data access, historical data access (trending and logging), and for alarms and events. Many times, these functions are implemented in a single server or client application. Because not all forms of data are supported by OPC the configuration and network analysis tools must support the protocol and access the devices directly. OPC lets any software read and write parameters, but for download of the control strategy and device configuration the software must fully support the protocol. The OPC clients run in the various workstations for operation, engineering, and maintenance. For medium and large systems, the OPC servers may run in a dedicated computer, whereas for small systems a server may run in one of the workstations.

Remote Access Server

For certain applications remote access to the network is necessary. In a remote application with an unmanned control system it may be required to dial into the system from a central location from time to time in order to check the status and download production information. This may be the case for oil fields or reservoirs, for example. Another application may enable the supplier or a plant engineer not on duty to dial into the system from any remote location to assist in troubleshooting. A third application is to connect two widely separated LANs together over the telephone network (figure 3-67).

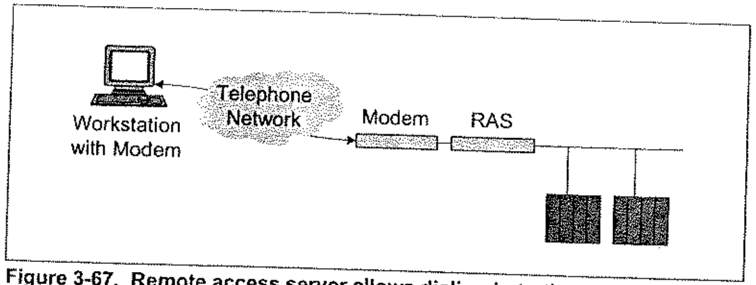


Figure 3-67. Remote access server allows dialing in to the network from a central location.

RAS can either be software running in a computer or a simple maintenance-free embedded device (figure 3-68). The RAS sits on the Ethernet and connects to the telephone network through a modem.

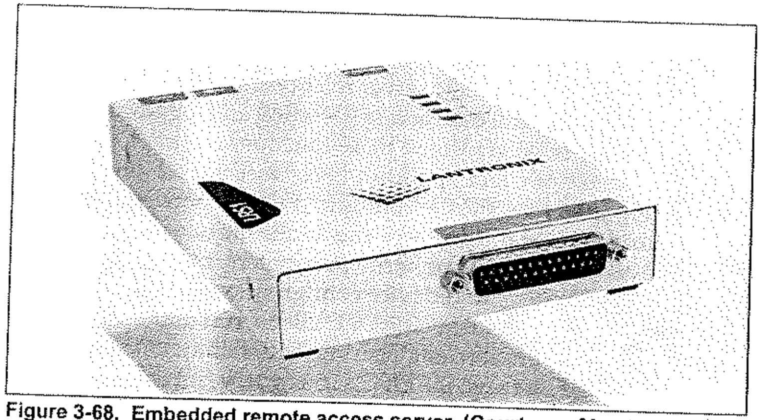


Figure 3-68. Embedded remote access server. (Courtesy of Lantronix)

The RAS has built-in optional policies for which protocols can be used as well as dial-back capability to a predefined number to control access.

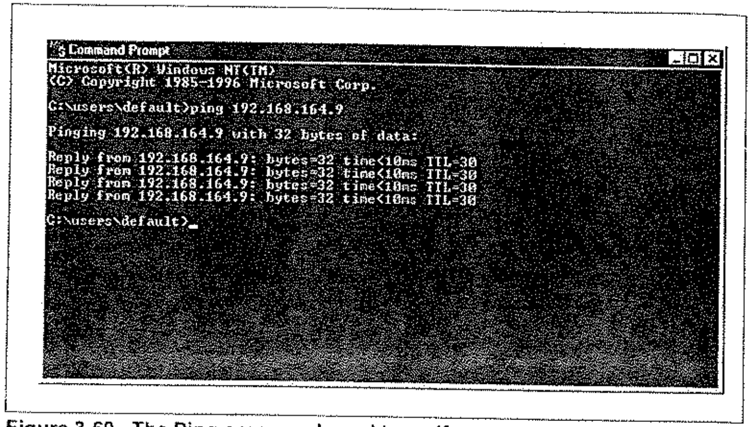
IP Commissioning

For IP networks, the address must either be configured manually before the device or workstation is connected to the network, or it must be assigned automatically by a DHCP (dynamic host configuration protocol) server on the network. Most companies have a MIS department that locally administers network addresses within the company. It may be good idea to assign a person from the MIS department to the project team to help it reach agreement on how the control system will tie in with the rest of the company's IT infrastructure.

Once a device or workstation is installed, powered, and connected to the network you can check its basic operation using the Ping command (figure 3-69). If the ping times out, the device or computer is most likely connected to the wrong network, or the network parameters are configured wrongly. Another possibility is that the device is not wired or operating properly, and you will need to do some troubleshooting. The Ping command is thus an excellent tool for commissioning, since it is very easy to confirm whether the devices are wired properly. To ensure that a device has not been mixed up with another one on the same network, the operator can observe the activity LED on the device while pinging to confirm that the correct device is being accessed. You should also use the Ping command to check whether an IP address is already used, if you are unsure. Use the arp -a command to see all the nodes the computer talks to.

Subnet

It is a good idea to partition a large network using routers because this will increase performance in networks that have many bridges or lots of broadcast traffic. You can partition the host number portion of the IP network address into a subnetwork number portion and a smaller host number portion for each of the several smaller subnetworks, each having fewer devices. In any case, a subnet mask is used to point out which portion of the IP address is the network identifier and which (the remainder) is the host portion for device addresses. Every device must be configured with a subnet mask if you want it to be able to work out which particular subnet it is on and whether the devices it wants to communicate with are on the same or a different network.



```
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\users\default>ping 192.168.164.9

Pinging 192.168.164.9 with 32 bytes of data:

Reply from 192.168.164.9: bytes=32 time<10ms TTL=30
Reply from 192.168.164.9: bytes=32 time<10ms TTL=30
Reply from 192.168.164.9: bytes=32 time<10ms TTL=30
Reply from 192.168.164.9: bytes=32 time<10ms TTL=30

C:\users\default>_
```

Figure 3-69. The Ping command used to verify any IP device even before control system software has been loaded.

For example, if the private network identifier 192.168.0.0 is used for the overall plant network, approximately 65,536 devices could be connected on one single network. That is far too many for efficient communication. Instead, the network can be partitioned into 256 subnets, each having a more reasonable 256 devices. Perhaps only one of those subnets will be used at first.

Generally, four numbers separated by dots—just like the IP address itself—represent a subnet mask. Construct the subnet mask in binary format by setting bits to 1 to indicate the network portion and to 0 to indicate the host identifier portion.

EXAMPLE 3-9

For example, to utilize the private address space from 172.16.0.0 to 172.31.255.255 for one network you should set the subnet mask host identifier to the first twelve bits:

Start address: 10101100.00010000.00000000.00000000

End address: 10101100.00011111.11111111.11111111

Subnet mask: 11111111.11110000.00000000.00000000

Subnet mask in decimal: 255.240.0.0

The 255.240.0.0 subnet mask is then configured in all the devices on the subnet.

For simplicity, it is highly recommended that you use subnet masks that are based on the full byte, using either 0 or 255, that is, 255.255.255.0 or 255.255.0.0 or 255.0.0.0.

When a device has a message to transmit it can do so directly if the destination network address is on a local network. Otherwise, the message must go through a router to reach the distant network. Using its configured subnet mask the device figures out which subnet it is on and which subnet the destination device is on in order to determine whether they are the same. If there is a router, the device must also be configured with the address for the router so it will know where to send the messages destined for other networks.

EXAMPLE 3-10

For example, in the network configuration in figure 3-70 two subnets are formed using the subnet mask 255.255.255.0. The control system subnetwork number is 192.168.164.0, and the office subnetwork number is 192.168.165.0. The other possible subnets are unused. If the workstation with address 192.168.165.102 wants to send a message to the linking device with address 192.168.164.100, it will compare its own (source) and the destination address against the subnet mask using bitwise AND:

Source address: 11100000.10101000.10100101.01100110

Subnet mask: 11111111.11111111.11111111.00000000

AND: 11100000.10101000.10100101.00000000

Source subnet: 192.168.165.0

Destination address: 11100000.10101000.10100100.01100100

Subnet mask: 11111111.11111111.11111111.00000000

AND: 11100000.10101000.10100100.00000000

Destination subnet: 192.168.164.0

Since the source and destination subnets are different, the workstation knows it must send the message to the router for onward forwarding to the 192.168.164.0 network in order for the message to reach the linking device.

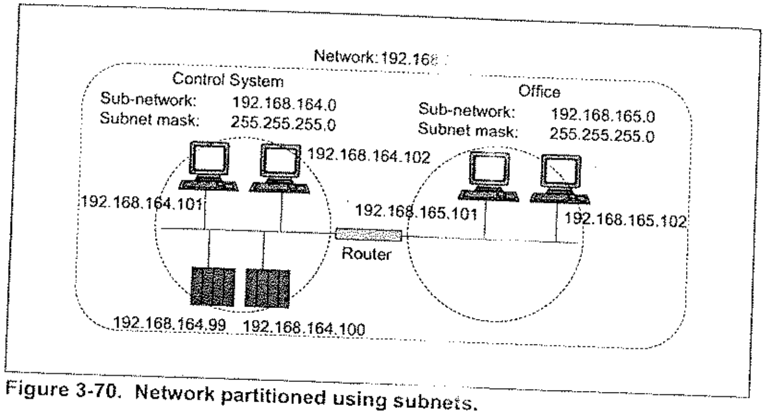


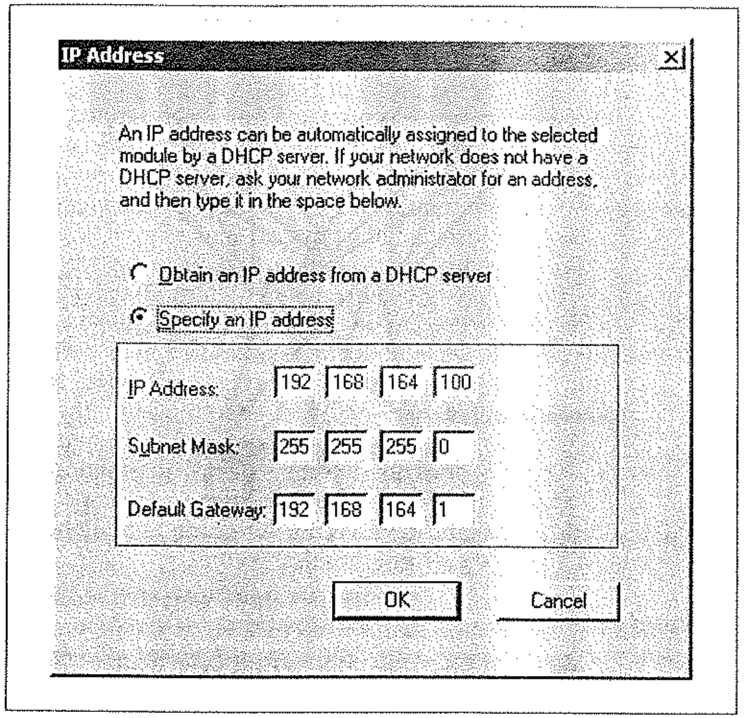
Figure 3-70. Network partitioned using subnets.

It may be helpful to think of the IP address as a phone number with the network identifier as the area code assigned to the network the device is located on. If the destination device has a different area code then it has to make a long distance call. The subnet mask is like the dash that tells you where the area code ends and the phone number starts in the long string of digits that constitute a phone number.

Manual IP Configuration

Once the IP address and subnet mask have been worked out they can be set in the devices and workstations. If there is a router on the network to link to other networks, you should also configure the router IP address in the field that is typically called "Default Gateway." If there isn't, leave it blank (Figure 3-71). The host portion of the IP address cannot be set to all binary zeroes or all ones. For workstations, configuring a manual address is straightforward since the information can simply be keyed in. However, to be configured manually devices that have no local user interface usually need to be connected to a computer on a separate network that is either running a special configuration tool or to terminal software like Telnet. The IP address must be set before the device is connected to the operational system network to avoid disrupting the operational network. A manually assigned address is static.

The address must be unique on the network. If the address is not set correctly the device will not be detected and may even disrupt the communication of the devices already on the network. It is therefore important to take measures to prevent any duplication of addresses. Duplicate addresses may disrupt supervision or even control. It may therefore be a good idea to include a tag and



IP Address

An IP address can be automatically assigned to the selected module by a DHCP server. If your network does not have a DHCP server, ask your network administrator for an address, and then type it in the space below.

○ Obtain an IP address from a DHCP server

● Specify an IP address

IP Address: 192 168 164 100

Subnet Mask: 255 255 255 0

Default Gateway: 192 168 164 1

Figure 3-71. Linking device IP address configuration.

address cross-reference list for each network as part of the system documentation.

Automatic IP Configuration

Assigning addresses automatically is the easiest solution, especially for large networks. Automatic address assignment makes it possible for devices to simply be plugged into the network. To use it, the network must have a DHCP (dynamic host configuration protocol) server, which is software that runs on one of the computers in the network. A DHCP server automatically assigns unique IP addresses to nodes on the network. This address is dynamic and may change from time to time. All the user has to do is to ensure that the devices connected support the DHCP protocol and that they are configured to have the address assigned by a DHCP server rather than manually. Usually, devices are shipped ready to accept a DHCP-assigned address.

FOUNDATION HSE

Once an HSE device is connected to the Ethernet it will be detected automatically by the host. For FOUNDATION Fieldbus, the network configuration database is prepared from user-defined device tags without any association with particular physical devices. To shorten the project cycle it may be a good idea to start preparing the configuration immediately, even before the devices have been received. Once installed, the physical devices need to be associated with their respective configuration. Each FOUNDATION Fieldbus device has a thirty-two-character identifier (ID) that is unique in the world and is used to unambiguously distinguish one device from the others. An HSE device is associated to its particular configuration by correlating the device configuration tag to its unique ID. This is a simple point-and-click operation since the host automatically detects the IDs of all HSE devices on the network. If the engineering tool does not detect the device automatically the device is most likely connected to the wrong network. Another possibility is that the device is not wired or operating properly, and you will need to perform troubleshooting.

Redundancy

To be used in automation Ethernet has to be made "industrial strength." The host-level network ties the whole system together, linking the various subsystems to the host. Thus, the visibility of hundreds and perhaps thousands of loops depends on the host-level network, as does the visibility of some intra-area control loops. Because so much information is concentrated on one network a complete failure could result in heavy losses. High availability for the host-level network is therefore paramount. Even in the commercial world, there are instances where high availability is important; for example, servers for e-commerce must be available round the clock. Consequently, there are already several solutions for fault tolerance available that can also be applied in industry. There are three basic philosophies for the host-level network redundancy: media redundancy, complete network redundancy, and Ethernet device redundancy. These can be combined to achieve even higher fault tolerance. Additional measures to increase availability are discussed in chapter 10, "Availability and Safety."

Media Redundancy

Media redundancy works entirely on the physical media level and is therefore independent of protocol. Media redundancy can therefore be used with any protocol that travels on Ethernet. Dual-path media redundancy can be achieved by forming a single-ring topology. The two paths use regular UTP or fiber media, but the active components have built-in intelligence to handle switchover. Regular nonredundant Ethernet devices can be used with these alternatives since the redundancy is managed by the media components that are transparent to the Ethernet devices, which need not take any action themselves. Thus, media redundancy only duplicates the networking itself; it does not manage a primary and secondary of a redundant pair of devices or communication ports. Simple media redundancy can be used when the chance of device failure is relatively small in comparison to the impact. Media redundancy alone provides sufficient availability for many applications.

One way to achieve media redundancy is to duplicate the entire star topology network by using redundant port transceivers near each device (figure 3-72). A redundant transceiver has three ports: one for the device itself, one for the primary communications network path, and another for the secondary communications path. Only one path is used at any one time. If the primary path loses its link, then the transceiver will immediately—in some cases, as quickly as a microsecond—switch to the secondary path. When the primary path reestablishes its link, the transceiver will automatically switch back to the primary path. In the unlikely event that the transceiver fails both paths are lost.

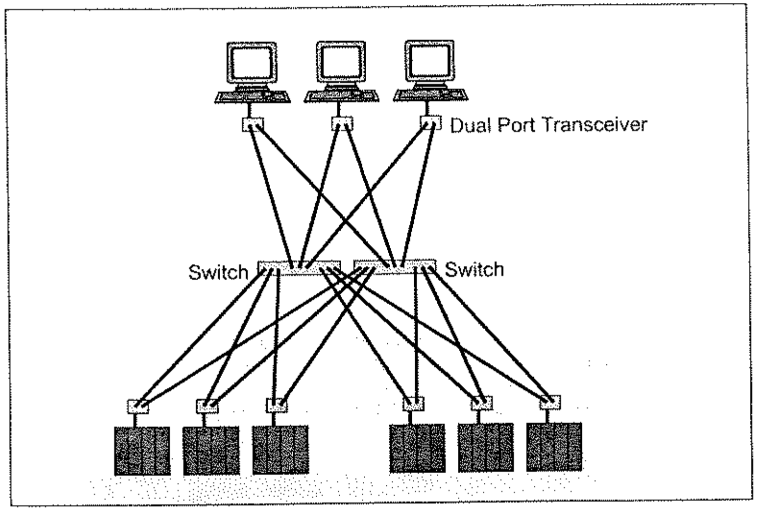


Figure 3-72. Media redundancy using dual port transceivers.

In a ring topology, special network hubs are connected in a physical ring formation (figure 3-73). A regular Ethernet device connects to the ring through these hubs, which appear like normal star topology hubs to the standard devices. The ring topology creates a dual communication path because the communication can go either clockwise or counterclockwise.

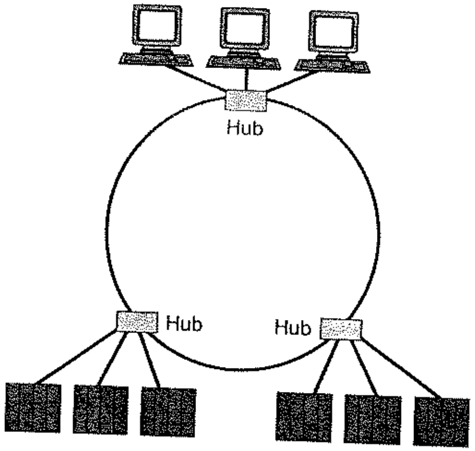


Figure 3-73. **Media redundancy using single ring topology.**

The distance from ring hub to device should be short to reduce the chances of damaging the nonredundant cable segment. To minimize the impact of a hub failure, it is also a good idea to use several small hubs that have only a few ports each rather than just a few larger central hubs (figure 3-74). Some solutions may require the use of a central unit to perform redundancy management, which may be a weak point in the system.

It should be noted that the ring topology is not part of the Ethernet standard, and all the hubs taking part in the ring topology must therefore be the same brand in order for the ring to work. However, the proprietary aspects of the ring topology occur at a very low level and do not affect the protocols and applications because they communicate independently of whichever Ethernet topology was chosen. The wires and fibers used in the ring formation are electrically and mechanically compatible with the standard Ethernet star topology, so you can use standard cable and connectors to connect the ring hubs to each other.

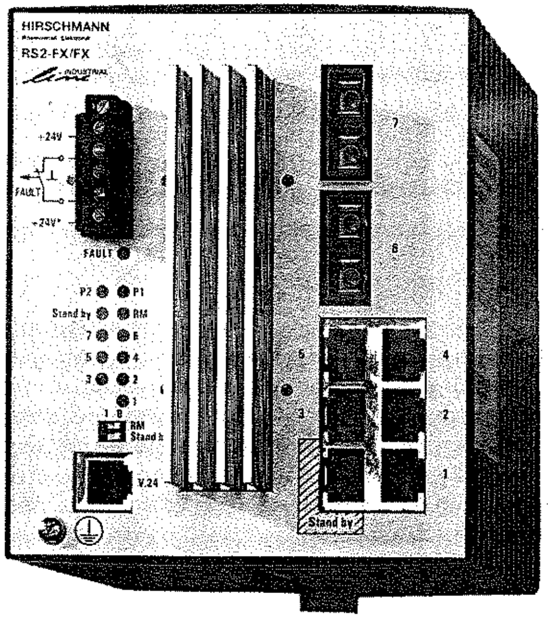


Figure 3-74. **Industrial grade fiber-optic Ethernet switch for ring-topology. (Courtesy of Hirschmann)**

Device and Port Redundancy

Host-level network devices like linking devices and possibly controllers marshal information for many loops. They are therefore relatively centralized compared to the highly distributed field-level devices, which often only handle a single point. This aspect makes them more critical to the operation of the system. Generally, it is a good idea to use completely redundant device pairs or perhaps devices with redundant communication ports whenever the need for high overall availability exists (figure 3-75).

Device and port redundancy works on a higher level than media redundancy. For this reason, the protocol must have network diagnostics and other functionality to select which device in a redundant pair and which redundant port a transmitting device should address. A redundant device pair consists of a primary and secondary device. If you use redundant device pairs with single ports you should connect the port of the primary device to the primary network. Likewise, the port of the secondary device in the pair should be connected to the secondary network. If you use redun-

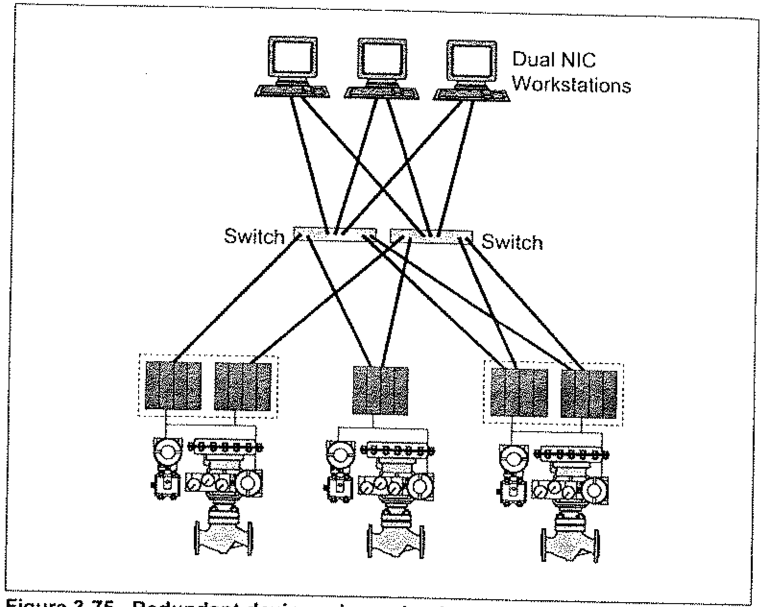


Figure 3-75. Redundant device pairs and redundant communication ports.

dant pairs of dual-port devices both the primary and secondary devices should be connected to the primary and secondary network. You must fit the workstations that connect to both the primary and secondary networks with two network interface cards or a single dual-port NIC.

Because the functionality of device and port selection is protocol dependent only devices that use the same protocol can participate in the redundancy scheme. Device and port redundancy is recommended for systems where very high availability is required because they eliminate transceivers, common hubs, redundancy management devices, and other common points of failure.

FOUNDATION Fieldbus HSE Redundancy

Port and device redundancy configurations are possible using HSE media. Both a primary and secondary linking device in a redundant pair may be connected to the same field-level network to provide two complete separate communication paths from the H1 Fieldbus to the host. In the event of a fault along the primary path, data is channeled through on the secondary path. This ensures that the plant-floor data reaches the operator even if one linking device and one network fails—thus, there is always a window to the process. Redundant devices and ports should all be given different IP addresses. In a redundant pair, only the primary device is active, while the data in the secondary is kept synchronized. If the primary fails, the secondary takes over the operation without any exchange of addresses. Thus, the other devices in the system see only a single device with one single tag but different IPs depending on which one is active. The HSE protocol in redundant devices makes sure the devices transmit using the correct address. Since the device and port redundancy is protocol dependent, all devices should be HSE. Non-HSE devices can only benefit from media redundancy. HSE devices are generally available with several options for redundancy.

The exchange of redundancy management information is part of the standard protocol. As a result, devices from different manufacturers can participate in the redundancy on the host-level network. However, to achieve the quickest switchover in the event of device fault you should ensure that the two devices in a redundant pair are identical and therefore from the same manufacturer. The primary and secondary should be mounted some distance apart to eliminate common mode failures. Only a single configuration has to be prepared, and when it is downloaded it goes to the primary. The secondary is then synchronized automatically, which makes the primary and secondary appear as one single device. In other words, from the point of view of control strategy and operations the secondary is not seen. However, from the point of view of device diagnostics both the primary and secondary are separate entities that can be individually diagnosed and checked based on their different addresses. Doing so makes it possible to ensure that the secondary device is also fully functional and ready to take over at any moment. That is, the operations only see the active devices, whereas the diagnostics sees both the active and passive device.

EXERCISES

3.1 What is the minimum impedance for a HART network?

3.2 Is it sufficient to install surge protection on the field devices?

3.3 Which is the point-to-point polling address used by HART?

3.4 What is the difference between a network, a segment, and a link?

3.5 Is preventing reflection the only function of a terminator?

3.6 Is ring topology supported by IEC 61158-2?

3.7 Is type A or B cable mandatory for IEC 61158-2 installation?

3.8 Can the spurs on a network of thirteen spurs exceed 90 meters in length?

3.9 Should the negative of the fieldbus wire be grounded?

3.10 What is the difference between a link, a linking device, and a coupler?

3.11 For a traditional entity barrier that provides 60 mA, how many devices can be connected, assuming each consumes 12 mA?

3.12 Does a simple terminator have to be certified as intrinsically safe?

3.13 Is a FISCO barrier linear or trapezoidal?

3.14 Can more than one barrier be connected to one communication interface port?

3.15 How many devices connect to a 10Base-T segment?

3.16 Which type of hub should be used to achieve best performance, a share or a switched?

3.17 How many switching hubs can be connected between two devices?

3.18 Which kind of device is used to partition large networks, a gateway or a router?

3.19 Does media redundancy alone increase availability?

BIBLIOGRAPHY

1. Magison, E. C. *Electrical Instruments in Hazardous Locations*, 4th ed., ISA-The Instrumentation, Systems, and Automation Society, RTP, NC, 1998.
2. Fieldbus Foundation, *31.25 kbit/s Intrinsically Safe Systems, Application Guide AG-163*. Fieldbus Foundation, Austin, TX, 1996.
3. Fieldbus Foundation, *Wiring and Installation 31.25 kbit/s, Voltage Mode, Wire Medium, Application Guide AG-140*, Fieldbus Foundation, Austin, TX, 1996.
4. Johannesmeyer, Ulrich. *Investigations into the Intrinsic Safety of Field Bus Systems*, PTB report W-53e, PTB, Braunschweig, Germany, 1994.
5. *PROFIBUS Draft Technical Guideline, PROFIBUS-PA User and Installation Guideline*, version 1.1, September 1996, order no. 2.092 JB: PROFIBUS Nutzerorganisation e.V., Karlsruhe, Germany.

4

Configuration

In a modern system that uses fieldbus the engineer must configure the network as well as the control strategy and the devices. Because the configuration of the network, devices, and control strategy are intimately related to each other it may be beneficial for you to do all of them at once using a single integrated engineering tool (figure 4-1). The same tool may also handle diagnostics and calibration as well as other maintenance and asset management functions, which are discussed in chapter 9, "Maintenance and Asset Management." Typically, the configuration tool will allow you to make strategy changes on or off line. "Offline" means that configuration changes are only stored in the database, not in the devices. Once the offline configuration is complete it can be downloaded to the devices. Since online changes are made to the devices directly, for safety reasons online configuration is usually restricted to changing parameter values. During offline configuration, the plant operators can change any parameter that is permitted to be written regardless of the set block mode. However, for online configuration changing certain parameters is not safe. For these parameters, in online configuration you must change the mode of the block before you modify the parameter in order to avoid a process upset. Configuration is typically done off line initially, during the engineering stage, but once the plant is operating minor adjustments are usually done on line.

Network configuration includes the allocation of networks to communication ports and linking devices as well as the setting of communication parameters. Device configuration includes selecting devices and setting actuator type, sensor type, making connections, and the like. Configuring control strategy includes linking

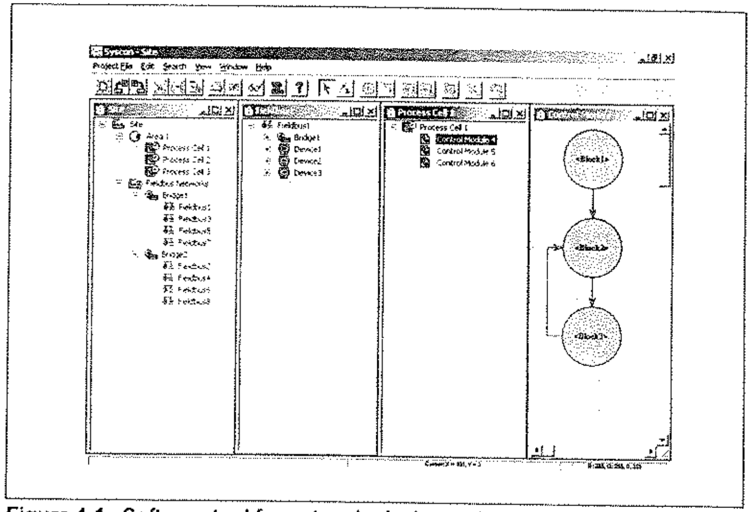


Figure 4-1. Software tool for network, device and strategy configuration.

and configuring the function blocks. Because HART networks are usually stand-alone without a direct link to the host-level network configuration is usually not required.

FOUNDATION Fieldbus specifies a graphical control strategy programming language that is based on the function block diagram concept. PROFIBUS PA and HART do not specify any programming language, and therefore device configuration is done separately from the strategy configuration. In systems that use PROFIBUS and HART control is typically done in a centralized controller, and the strategy configuration language and strategy configuration tool are unique to whichever controller is used. Whether the system configuration is simpler when device configuration is separate from or integrated with the control strategy configuration is up to the individual user's preference. PROFIBUS PA has function blocks with parameters that are very similar to those of FOUNDATION Fieldbus but with an important difference: PROFIBUS PA function blocks cannot be linked. Because of their common heritage the parameters in FOUNDATION Fieldbus and PROFIBUS PA are very similar.

HART devices are usually configured using a portable handheld terminal, but PC software is also available. There are also remote terminal units (RTU) and calibrators that communicate with HART instruments. HART devices are usually only configured on line, that is, while connected to the device, although a handheld usually has the capability of preparing and storing a configuration off line in advance for later download. Because of their conceptual similarities there are software applications on the market that support both HART and PROFIBUS PA in the same single tool.

FOUNDATION Fieldbus and PROFIBUS devices are configured using PC software that is permanently connected to all the field instruments. To shorten the project time it is recommended that you use a software tool that supports offline configuration. In offline mode, the software tool is not connected to any device. The configuration for all the instruments in the plant is prepared in advance, even before the devices are manufactured, delivered, or installed. The configuration is stored in a file, and once the devices are commissioned the configuration can be downloaded. Actually, a large number of settings must be made in the devices to make the communication work, but the configuration tool typically handles this automatically. Parameterization is typically done on line in a plant once it is up and running.

Network Configuration

As far as the user is concerned network configuration is a matter of defining the network hierarchy for the host-level and field-level networks. HART is typically used as a point-to-point topology on its own whereas FOUNDATION Fieldbus H1 and PROFIBUS PA networks generally connect to the host-level network using linking devices to form sophisticated network architectures. In other words, for HART there is not really any network configuration to be performed. For both FOUNDATION Fieldbus and PROFIBUS PA some quite sophisticated communication configuration must be performed, but it is generally done automatically by the configuration tool and is therefore not a matter of concern to the user. These unseen details are explained in chapter 11, "How Fieldbus Works."

Assigning addresses in HART and PROFIBUS PA is explained as part of device commissioning in chapter 3, "Installation and Commissioning."

To make it easy to cross-reference between the network configuration and the electrical diagram for the network, it may be a good idea to enter the cable name and loop diagram drawing number in the configuration tool.

PROFIBUS

From a communications perspective, there are three types of devices on the PROFIBUS network: slave, class 1 master, and class 2 master. A class 2 master is generally a configuration tool (figure 4-2). A class 1 master is typically a centralized controller like a PLC. The class 1 master cyclically exchanges I/O information with the slaves. A class 2 master acyclically exchanges configuration and maintenance information in the slaves.

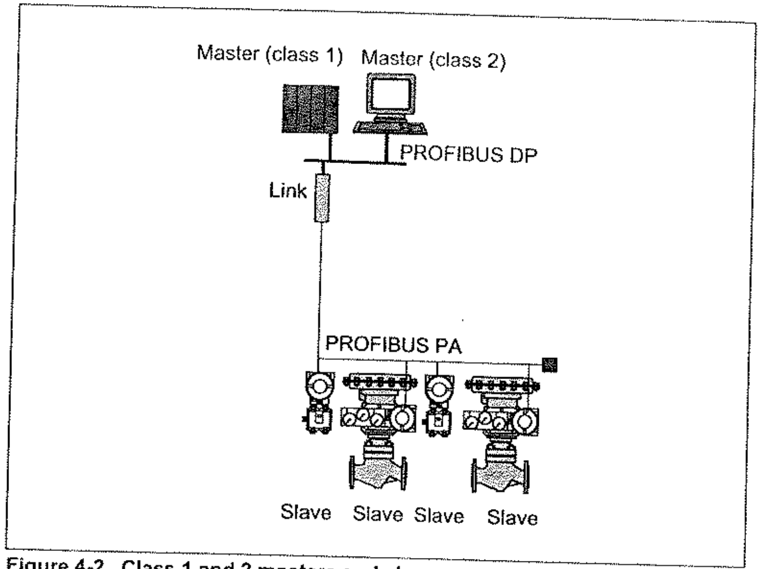


Figure 4-2. Class 1 and 2 masters and slaves.

The masters are generally connected at the PROFIBUS DP host-level network, and they access the PROFIBUS PA devices through a linking device or possibly a coupler.

FOUNDATION Fieldbus

From a communications perspective there are three types of devices on an H1 network: basic, link master, and bridge (figure 4-3). The host interface is generally a linking device that is connected on the host-level HSE network. Workstations access data through the linking devices. Some devices may support two or more of these functions and can be configured under the heading BOF class to either one of the supported roles. Only one of the link masters takes on the role of link active scheduler (LAS) at any one time. The functioning link masters automatically negotiate who will be the LAS. If the current LAS fails another link master takes over automatically.

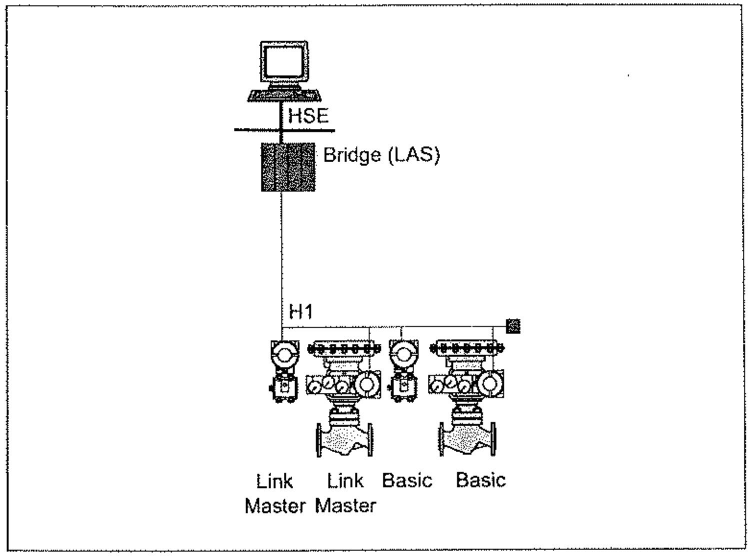


Figure 4-3. Bridge, basic and link master devices out of one is the LAS.

The LAS performs several functions, including detecting new devices and assigning their node address, synchronizing time among devices, and controlling data transfer. For high availability it is a good idea to use redundant linking devices. If your system uses only a single linking device for the network a backup LAS in a field device is another option. Most field instruments are generally configured as a basic device. Generally, it is preferable to use the host interface, such as a linking device, as the primary LAS. If you use redundant linking devices the secondary linking device takes over the LAS function should the primary device fail. Additionally, you may configure some field devices as link masters, which allows them to take on the LAS role in case no functioning linking device is present on the network. This is commonly referred to as a backup LAS. It may be desirable to use a backup LAS to achieve greater availability if you don't use redundant host interfaces. A bridge passes data from one H1 network to another, which allows devices on different networks to communicate. Bridging is functionality that is generally built into a linking device. As part of the network configuration the engineer must assign the preferred link master and application clock time publishers. A bridge must be configured as the primary link master. By default, the primary LAS would generally be the host interface. The host auto-detects and identifies new devices on H1 and HSE.

Device Configuration

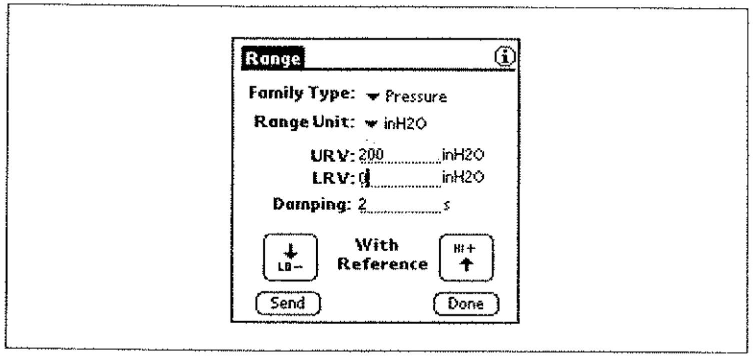
HART, FOUNDATION Fieldbus, and PROFIBUS PA all cater very well to device configuration. These three protocols all have standardized parameters for configuring the devices' operation, a common characteristic that sets them apart from other protocols. These are therefore the most common protocols for field instruments. The parameters for PROFIBUS PA and FOUNDATION Fieldbus are organized in blocks. It is a good idea to use templates that have device configurations prepared and verified in advance. Doing so makes configuration faster and reduces mistakes.

HART Devices

Functions in the HART protocol correspond to three classes of commands: universal, common practice, and specific. Every HART device supports all the functions associated with the universal commands as well as some of the functions associated with the common practice commands. The common practice functions are implemented consistently in all devices. For the most part, all HART devices are therefore operated in the same basic way, and any HART handheld can configure most device functions even if it hasn't been loaded with the device description for the device. To access all functions, the manufacturer must either tailor the handheld application for the device or be able to interpret the device description file.

HART provides for basic device identification. Although the identification information does not affect the operation of the device it is still a good idea to set it as part of device configuration process as it will be helpful during commissioning and maintenance. An eight-character physical device tag can be set, for example, "PIT-2308." The sixteen-character descriptor may be used to describe its application, for example, "Boiler 1 level." Special considerations or instructions can be stored in a thirty-two-character message, for example, "ladder needed for access." Lastly, a date can also be entered; it may be used to record the date of purchase, of installation, or of the last or next maintenance or calibration.

In a vast majority of cases, the 4-20 mA output of a HART transmitter is used, and therefore you must set a measuring range for the device. The lower range value (LRV) is the value at which the current output shall be 4 mA (0% of range), and the upper range value (URV) is the value at which the current output shall be 20 mA (100% of range). The difference between the URV and the LRV is called the span. LRV is frequently called zero. It is very important to configure the range of any displays, controllers, or host in the system to the corresponding values; otherwise, the indication will be false, and the control strategy will not function properly. It is therefore crucial that you carefully document and cross-reference the ranges between transmitters and displays and manage any changes to reduce inconsistencies. When you perform the transmitter reading using the digital communication the process value is displayed in engineering units independent of the range.



Family Type: ▾ Pressure
Range Unit: ▾ inH2O
URV: 200 inH2O
LRV: 0 inH2O
Damping: 2 s

Figure 4-4. Range setting.

Range setting can be done in two ways, to applied input or to a user-entered value. In the case of the applied input the LRV or URV is set to the applied input value at the press of a button on the handheld (figure 4-4). Usually, transmitters have some form of local zero and span switches that have the exact same function. Pressing the Set LRV button sets the LRV to the applied input value, which makes the output 4 mA for the particular input. The value for the URV is adjusted by the same amount, which keeps the span unchanged. For example, if the range originally is 0-560 mmH2O and zeroing is done when a pressure of 190 mmH2O is applied, the range becomes 190-750 mmH2O. These are called non-interactive zero and span adjustments. Pressing the Set URV button configures the upper range value to the applied input value without affecting the LRV. This makes the output 20 mA whenever the applied input is at this value.

For example, when you are using a differential pressure transmitter on a tank that is exactly empty or in a pipe in which the flow is zero, press the Set LRV button to the applied input and the output becomes 4 mA. The LRV has automatically been set to the applied input, which cancels out the pressure from wet leg and even any erroneous engineering unit reading. In other words, it is not necessary to know what the actual input is since the transmitter just makes sure that whenever it is at that value the output is 4 mA. Likewise, if the tank is just full press the URV button to make the output 20 mA. The URV button does not affect the zero.

The second method for setting the range is simply to type in the desired values from the handheld keyboard. If a particular range is required, just key it in regardless of what the present input is. The second form of re-ranging does not require you to know the present physical input, and it can be done remotely using the HART communication.

EXAMPLE 4-1

For example, the liquid in the tank in figure 4-5 has a density of 800 kg/m^3. The range shall be set to measure the level up to 0.56 meter above the level tap. The transmitter is mounted 0.19 meters below the level tap. The local gravity is 9.81 m/s^2. To cancel the hydrostatic pressure from the wet leg the range shall be set 1.49 to 5.89 kPa.

$$LRV = 0.19 \times 9.81 \times 800 = 1.49 kPa$$

$$URV = (0.19 + 0.56) \times 9.81 \times 800 = 5.89 kPa$$

Many changes made to the device configuration, such as range, sensor type, and transfer function, will result in an instant change of the current output. If no precautions were taken, this would be interpreted by a controller as a sudden process upset, which it would try to correct for. Although there is no real process upset, the controller would cause one trying by to correct the one it mistakenly perceives. Therefore, the operator should put the control loop in manual mode before making such a change. Most handheld terminals will display a message advising the operator to do so before the change is actually effectuated. Once the change has been made the operator is reminded to return the loop to automatic.

The operator can change the engineering unit without affecting output because the transmitter automatically recalculates the range values in accordance with the new engineering units.

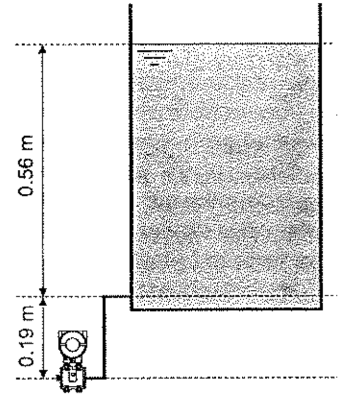
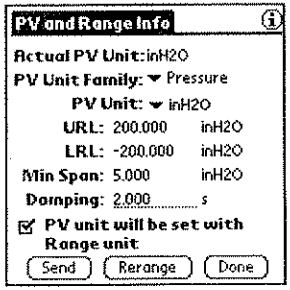


Figure 4-5. Level measurement with wet-leg.



PV and Range Info
Actual PV Unit: inH2O
PV Unit Family: ▾ Pressure
PV Unit: ▾ inH2O
URL: 200.000 inH2O
LRL: -200.000 inH2O
Min Span: 5.000 inH2O
Damping: 2.000 s
☑ PV unit will be set with Range unit

Figure 4-6. Sensor range limit information.

The operator can review the measurement range limits for the sensor (figure 4-6). This includes the upper range limit (URL) and the lower range limit (LRL) as well as the minimum span that the transmitter can accept. The minimum span is the smallest absolute difference between the URV and LRV that can be set. URL and LRL are the limits within which the URV and LRV can be set. Note that the operator can set URV and LRV to suit the application, whereas URL, LRL, and minimum span are fixed constants that depend on the sensor type. The minimum span is a limit imposed mainly to prevent misapplication, such as using a high-range transmitter to measure small values. This would result in poor accuracy. For both forms of range, when the user sets the transmitter it checks the range value the user is attempting to set against its limits to either accept it or reject it. Range violation messages will accordingly be displayed in the handheld. If the URL or LRL would be exceeded the technician must use a sensor module with a higher range. If the span would be too small the technician must use a lower-range sensor module.

Don't confuse range setting with calibration; they are two different things. Calibration means correcting the sensor reading to match the applied quantity. Setting the range means selecting at which values the current output shall be 4 mA (0%) and 20 mA (100%), respectively. Setting the range does not correct the sensor reading in the transmitter itself. The technician can set the range remotely without applying any input, but by definition calibration means that a known input from a standard has to be applied. For the HART protocol, calibration is usually called "trim." Do not use trim (calibration) to cancel out wet legs.

Current outputs below 4 mA or above 20 mA indicate that the measured value is out of the set range of the transmitter or that the sensor has failed. Most modern devices follow the NAMUR NE-43 standards, which dictate that the output can under-range and over-range from 3.8 mA to 20.5 mA; that is, 3.8-4.0 mA and 20.0-20.5 mA can be seen as "uncertain." The operator can set the transmitter fail-safe mode to bring the current output either high or low to ensure that the action in the controller is correct. For the low option, the current values in the range 3.6 to 3.8, and 20.5 to 21 mA indicate a "bad." High is typically the most suitable option since it exaggerates; for example, a high temperature or pressure levels automatically, which makes the controller take safe action. Some applications require a low fail-safe mode.

To filter out noise on the measurement signal you can apply damping. To adjust the damping you can configure the filter time constant, in seconds. Most devices have a write-protection lock to protect the configuration from being tampered with, typically by using an internal switch or jumper. It is possible to review the write-protection status from the handheld.

Pressure

Plants often use pressure transmitters for inferential measurements. For example, differential pressure transmitters are often used for flow measurements. For some flow applications, the plant uses differential pressure producers like orifice plates and pitot tubes and will desire a reading that is proportional to the volumetric flow. In this case, the engineer should configure the transfer function of the differential pressure transmitter input to

square root extraction. In most other applications the transfer function shall be linear. Some pressure transmitters may have other options for obtaining the volume of the tank, such as a look-up table for freely configurable linearization or square roots of the third and fifth powers used to measure open channel flow.

FOUNDATION Fieldbus Devices (H1 and HSE)

Configuring a device includes the process of entering the selection of the field-level and host-level devices that will be used and the configuration and parameterization of the resource block and transducer blocks. The engineer will give each device a physical device tag (PD_TAG) to identify it. The device tag must be unique in the system and can be up to thirty-two characters in length, for example, "PIT-2308." The physical device tag that is assigned should be the same tag that is used in the network diagram document. Although hosts generally auto-discover connected devices on both the H1 and HSE networks, this is only done when commissioning is done, after the devices have been installed. Since configuration generally starts long before the devices are manufactured the device selection is typically entered manually.

Blocks

The engineer sets parameters in the resource block and transducer blocks to get the desired device configuration. These blocks have many characteristics in common with the function blocks that are used to build the control strategy. The transducer, resource, and function blocks work the same way in H1 and HSE devices. Each device must be configured with one resource block, and all input and output devices can be configured with at least one transducer block, essentially one per measurement or actuation. Transducer blocks act as an interface between physical I/O hardware and function blocks. The resource block contains information about the device as a whole. Transducer blocks and the resource block have no input or output parameters and cannot be linked. Each block should be given a block tag (FB_TAG) to identify it. The tags that are assigned for function blocks should be the same tags as the ones used in the process and instrumentation (P&I) diagram. The block tag must be unique in the system and can be up to thirty-two characters in length.

It may be tricky to come up with tags for the plethora of blocks in a system. It may therefore be a good idea to make the resource block tag the same as the physical device tag and make the transducer block tags the *same as the associated input/output function block but with a suffix, for example, "PT-2308_T" (figure 4-7).*

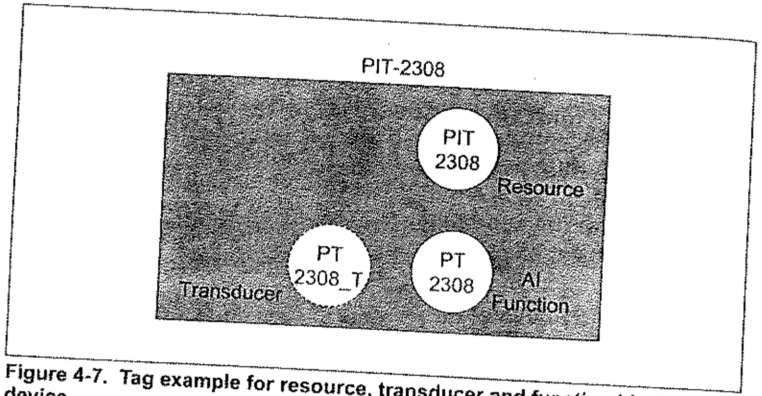


Figure 4-7. Tag example for resource, transducer and function block and device.

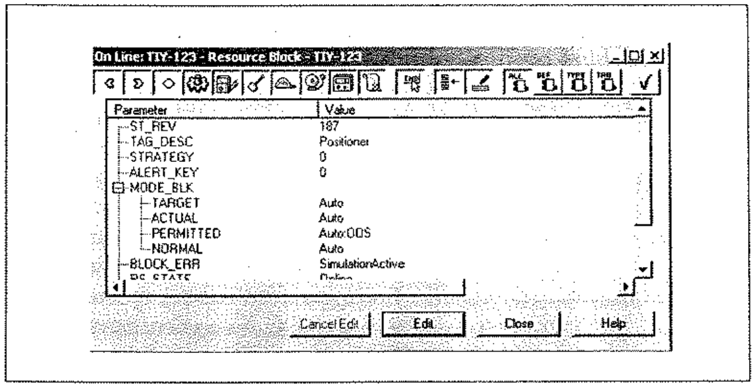
The standards include transducer blocks for several different device types. Each one has a set of parameters for common functions for the quantity measured or the form of actuation. Manufacturers also add an extended set of parameters to cater to the specifics of the sensing or actuation technology employed. Manufacturers may also have created blocks for the measurement of quantities or forms of actuation for which there is no recommendation in the specifications. However, the specification requires manufacturers to create blocks and parameters that are consistent with it. The specification also requires manufacturers to disclose these blocks and parameters using the DD so as to make interoperation in open systems possible and configuration intuitive. For the same reason, users should not try to learn every block by heart; rather, it is more important to understand the concepts.

The resource block is generally implemented without any additions. The transducer blocks are responsible for interfacing the physical sensors and actuators and therefore contain the detail diagnostics and calibration information. Setup that is related to I/O hardware, such as the selection and calibration sensor type and wiring, is done through parameterization in the transducer block that is associated with each input and output. This also makes it easy to check on the sensor measurement limits. Devices that have a local digital indicator may also have a display transducer, through which the display and local adjustment switches can be configured.

It is a good idea when building your control strategy to use the standard FOUNDATION function blocks as far as is possible. Use blocks that have proprietary extensions only when necessary. In other words, it is a good idea to use devices that support both the standard blocks as well as blocks with enhanced features. Devices that support dynamically instantiable function blocks can usually do this. This increases your chances of finding a matching replacement. It also makes it possible to "drag-and-drop" the block to another device if the replacement really does not have that block type.

Parameters

Each block contains a number of parameters, all of which have predefined names. The first six parameters are the same in every FOUNDATION block and are therefore called universal parameters. Many parameters are simple, that is, contain only a single piece of information, whereas some parameters are really a record of several elements, wherein each element is like a simple parameter containing one piece of information. For example, many parameters consist of a value and a status (figure 4-8). Many parameters for options are bit-enumerated, meaning that you can select several options in a list within the parameter. The parameters in the resource block and the transducer blocks are all "contained," meaning they cannot be linked. The parameter values are either set by the user or by the block itself.



Parameter	Value
ST_REV	187
TAG_DESC	Positioner
STRATEGY	0
ALERT_KEY	0
MODE_BLK	
TARGET	Auto
ACTUAL	Auto
PERMITTED	Auto:OOS
NORMAL	Auto
BLOCK_ERR	SimulationActive

Figure 4-8. Block parameters and parameter elements.

Some parameters can be both read and written, whereas other parameters can only be read. There are block parameters that affect the block output. Therefore, for safety reasons they can only be written when the block is in a certain mode so as to ensure that the process control is not upset by a bump in process variable or controller output. Parameters are stored in one of three different ways:

- Static storage
- Dynamic storage
- Nonvolatile dynamic storage

Static parameters do not change by themselves; either they are constant or it is the user who configures them, and the number of changes are counted. Static parameters are retained during power loss. Static parameters that cannot be written by the user are constants set by the manufacturer. Most contained parameters are static since they hold part of the device and strategy configuration. Dynamic parameters continuously change and therefore do not retain their value during a power loss. The process value is a common example of a dynamic variable. Nonvolatile parameters are like the dynamic parameters in the sense that they are typically updated continuously, but like the static parameters they remain constant in certain modes, and the last value is retained during a power loss. Setpoint and output are examples of nonvolatile parameters.

Parameters may be floating-point or discrete. Floats are generally in engineering units rather than scaled into percentages, although percentage is typically used for control output. The range for a float is from as small as $+/- 1.2 \cdot 10^{-38}$ to as large as $+/- 3.4 \cdot 10^{38}$. For certain parameters, floats can be configured as negative infinite (-Inf) and positive infinite (+Inf) and used, for example, to disable certain functions like alarms and integral action. Discrete parameters have 256 valid enumerated states, which makes possible not only "true" or "false" but many others, such as "open," "close," and "stop," depending on the block type.

Each device contains a tremendous amount of information, and it is therefore easy to get swamped by data. Locating the correct information can be like finding a needle in a haystack. The FOUNDATION parameters are categorized, and generally the configuration tool has filters to hide the information that is irrelevant for the moment:

- Input, Output, Contained
- Dynamic
- Diagnostic, Service, Operate, Alarm, Tune and Local

Universal Parameters

The tag descriptor (TAG_DESC) parameter may be used to describe the function the block is performing as a form of documentation that facilitates future configuration editing. The description may be up to thirty-two characters in length. The tag descriptor in the resource block may be used to document the function or location of the device as a whole, for example, "Boiler 1 drum level." The tag descriptor in a function block can be used to explain the function of the block, for example, "calculate setpoint from wildflow." This parameter does not affect the operation.

The strategy (STRATEGY) parameter may be used to help group blocks by giving a number in the range of 0 to 65535. It does not affect the operation of the block. To help identify the location of the block this parameter may be configured with a unique number, for example, for the control module as defined in ANSI/ISA-88.01-1995, Batch Control Part 1: Models and Terminology.

The number of writes of the configuration, that is, the revision level of the static parameters, is tracked by the static revision parameter (ST_REV). This parameter is incremented every time a change is made to a static parameter, that is, the total number of changes to the block. This parameter can only be read, not changed by the operator. It may be used to monitor changes in order to detect tampering with the block. This parameter is ideal for use by asset management software to keep track of changes to the block in order to establish an audit trail.

To help sorting or directing alarms and block events the alert key (ALERT_KEY) parameter may be configured with a number between 0 and 255. The parameter has no effect on the block operation. The alert key can be used for grouping associated alarms. For example, it can be used for "First Out" alarming in the host, that is, annunciating when the first alarm in the group occurs but annunciating differently for subsequent alarms in the group. This prevents "alarm avalanches" [see ANSI/ISA-18.1-1979 (R1992) - Annunciator Sequences and Specifications]. Alternatively, the alert key parameter can be used to help identify the location of an alarm. This parameter may be configured with a unique number, for example, for process cell or unit as defined in the ANSI/ISA-88.01-1995 standard.

A summary of the current hardware and software error status is kept in the block error (BLOCK_ERR) parameter. This parameter can only be read. The block error parameter is ideal for diagnostic overview and is used by asset management software to check health of a device. The following errors are traced:

- Block configuration error
- Link configuration error
- Simulate is active (enabled)
- In Local Override (LO) mode
- Device fault state is forced
- Device needs maintenance soon
- Input failure
- Output failure
- Memory failure
- Lost static data
- Lost nonvolatile data
- Readback check failed
- Device needs maintenance now
- Powering up
- In Out-of-Service (OOS) mode
- Others

"Simulation is active" is only seen in output and input class blocks. In the resource block, it means that simulation has been enabled by the hardware protection, if there is any. "Others" means that more status is available, so the technician should check the detail diagnostics parameters, for example, XD_ERROR in the transducer blocks.

Block Mode Parameter

The most important parameter in any block is the block mode (MODE_BLK) parameter. The mode of the block decides how the block is operating. There are eight different modes, but only two of those modes are really used for the resource block and the transducer blocks. Therefore, the Out-of-Service (OOS) and Automatic (Auto) modes are discussed first, and the other modes are discussed in the section on the FOUNDATION Fieldbus programming language. Traditionally, mode has only been set for the PID function, but in fieldbus the concept has been adopted in all blocks. Automatic is the normal operational mode for a transducer and resource block. In OOS mode the block is not operational. There are transducer block parameters that when changed affect the sensor reading or the position of the actuator. Therefore, for safety reasons during online configuration some parameters can only be written when the block is in OOS mode. This ensures that the process control is not upset by a bump in the process variable or controller output. During offline configuration there is no such restriction. The block mode parameter has four elements:

- Actual mode
- Target mode
- Permitted mode
- Normal mode

The desired operational mode for the resource block and transducers is automatic (Auto), and it should be set at the block mode "Target." Under certain conditions, this mode may not be attainable. The prevailing mode can be verified from the block mode "Actual" element. If the "Actual" mode differs from the target the reason can usually be found in the block error parameter. The other mode elements are primarily used by the FOUNDATION function blocks and are described in a later section of this chapter. The mode of the resource block controls the mode of the other blocks in the device (figure 4-9). If the resource block is set to OOS mode, all other blocks also go into OOS mode. In other words, putting the resource block in OOS mode is a method for putting the entire device into a "hold" state.

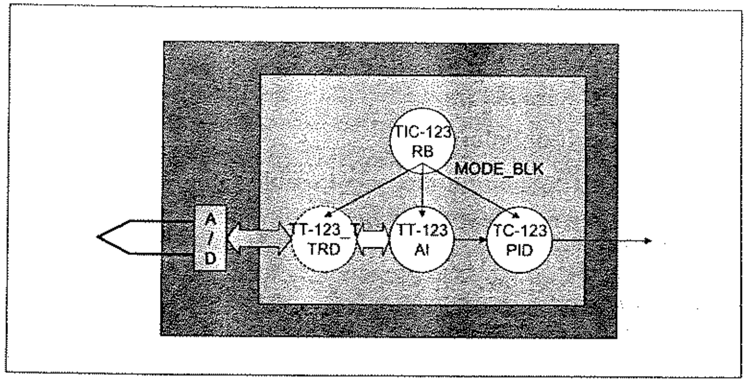


Figure 4-9. The resource block mode controls the mode of all blocks in the device.

Parameter Status

The measured or processed parameters in the transducer and function blocks consist of two elements: value and status. The status element contains additional validation information about the value. It is very helpful to refer to the status of parameters when troubleshooting since status indicates hardware, communication, or other fault. This makes it easier to trace back to the source of a problem such as sensor failure. The status element has three parts:

- Quality
- Quality substatus
- Limit condition

"Quality" is the general validity of the value: "Good," "Uncertain," or "Bad." The quality substatus explains the problem with a finer granularity. "Good" means that the value may be used for control. Depending on whether or not the block output can be part of a cascade structure, the "Good" quality is either "Good (cascade)" or "Good (noncascade)." The concept of cascade structure and the "Good" subquality is explained in detail in the section on the FOUNDATION programming language. In the transducer blocks the subquality for "Good" generally is "Nonspecific." "Bad" quality means that the value is not correct and should not be used for control. The subquality for "Bad" can assume the following reasons:

- Nonspecific
- Configuration error
- Not connected
- Device failure
- Sensor failure
- No communication - last usable value
- No communication - with no usable value
- Out of Service

The "Bad" subquality "Configuration error" means that some parameter is out of the valid range, an invalid option has been selected, or the parameter has not been configured at all. "Sensor failure" means that the associated sensor has failed. "Device failure" typically means that the associated output hardware has failed. If the block is in OOS mode, the subquality will be "Out of Service."

Some of the "Bad" subqualities will only be seen in function blocks because they are associated with links and processed inputs. "Nonspecific" is seen on outputs and is generally an indication that the function block has a "Bad" input. "Not connected" is seen on inputs and means that the input must be connected but isn't. "No communication" is seen on inputs and means that the output from the connected function block is not being received because the communication is not working. Depending on what the communication problem is the subquality can be either "No communication - last usable value," which means that communication has been working since the block was last in OOS mode but has now failed, or it can be "No communication - with no usable value," which means that communication has not been working since the block was last in OOS mode.

The "Uncertain" quality is associated with measurement values and means that the value is questionable and may not be totally correct. The problem is less severe than the problems associated with "Bad." Function blocks offer the option of either continuing to operate using the associated value in order to provide a high availability or rejecting the value as a safety precaution. The subquality for "Uncertain" can involve the following reasons:

- Nonspecific
- Last Usable Value (input)
- Substitute (input)
- Initial Value (input)
- Sensor Conversion not Accurate
- Engineering Unit Range Violation
- Subnormal

The "Uncertain" subquality "Sensor Conversion not Accurate" means that the measurement has reached either nominal limit or is degraded due to, for example, fouling. "Engineering Unit Range Violation" means that the value has exceeded its range. "Subnormal" means that one of many sources for a derived value is not good. For example, with dual redundant inputs "Subnormal" would indicate for 2oo2 that one input is degraded and the value is based on one remaining input.

Some of the "Uncertain" subqualities will only be seen in function blocks because they are associated with links and processed inputs. "Nonspecific" is seen on outputs and is generally an indication that the function block has an "Uncertain" input. "Last Usable Value" is seen on an input parameter and means that the configuration tool disconnected the link. "Initial Value" is seen on an input parameter and means that the value was entered while the block was in OOS mode.

Resource Block

One resource block must be configured per device. The resource block contains information that is common to the device as a whole. All the parameters are contained in the block, that is, they cannot be linked (figure 4-10). The block is responsible for the device's overall diagnostics. Setting Out-of- Service mode in the resource block stops all function blocks in the device. Automatic is the normal mode of a resource block.

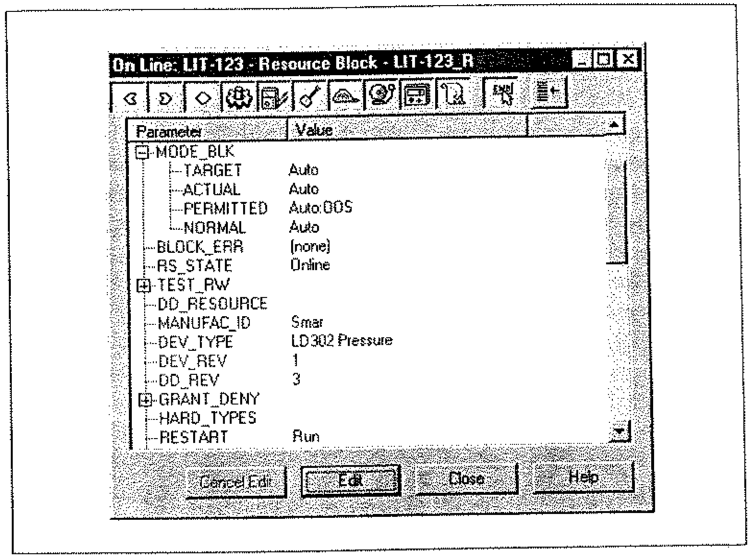
There are a number of parameters for identifying the device. The manufacturer of the device is identified by the manufacturer identification (MANUFAC_ID) parameter. The device type (DEV_TYPE) parameter identifies the basic device model. The revision of the device and the device description is identified by the device revision (DEV_REV) and DD revision (DD_REV) parameters, respectively. If the device description is stored in the device, the name of the resource that contains the device description (DD) is indicated in the DD resource name parameter (DD_RESOURCE). If the device description is not stored in the device the name is blank. This parameter is not useful to humans but may be helpful for the host device. The interoperability test kit version (ITK_VER) parameter indicates the device and DD revision for the test kit that was used in the registration test of the device.

The information about which input and output hardware the device has is kept in the hardware types (HARD_TYPES) parameter:

- Scalar input (AI)
- Scalar output (AO)
- Discrete input (DI)
- Discrete output (DO)

The resource state (RS_STATE) parameter reflects the status of the part of the control strategy that is in the device. There are six different resource states:

- Start/Restart
- Initialization
- Online
- Online Linking
- Failure
- Standby



Parameter	Value
MODE_BLK	
TARGET	Auto
ACTUAL	Auto
PERMITTED	Auto:OOS
NORMAL	Auto
BLOCK_ERR	(none)
RS_STATE	Online
TEST_RW	
DD_RESOURCE	
MANUFAC_ID	Smar
DEV_TYPE	LD302 Pressure
DEV_REV	1
DD_REV	3
GRANT_DENY	
HARD_TYPES	
RESTART	Run

Figure 4-10. Resource block parameters are contained.

If the resource block mode is Auto the resource state is usually "Online" by the time the user gets to see it. In OOS mode, the resource state is "Standby." If the resource state is "Failure" it means that a memory or other hardware failure has been detected.

The report of any alarm in the device is sent to the host. If it is not confirmed as being received by the host within the time specified by the Confirm time parameter (CONFIRM_TIME) the alert will be re-sent to the host. This parameter is configured in units of 1/32 of a millisecond. That is, for a confirmation time-out of 30 seconds you should set the value as 960,000. A maximum number of unconfirmed alert notifications can be queued in a device. This maximum is indicated by the max notify parameter (MAX_NOTIFY). The user may set a limit lower than this number by adjusting the notification limit parameter (LIM_NOTIFY).

In case parameters classified as nonvolatile are not stored in nonvolatile memory their values are copied to nonvolatile memory with an interval that is set in the nonvolatile cycle time (NV_CYCLE_T) parameter. If the time is set to zero, variables will not be copied and will therefore not be volatile. This parameter is configured in units of 1/32 of a millisecond just like the report confirmation time. Independent of the cycle time configuration, values entered by the user are always copied to nonvolatile memory.

Through the restart parameter (RESTART), there are three ways to restart the device:

- Resource
- Uninitialized
- Processor

Selecting the "Resource" option restarts the execution of the control strategy without changing the configuration. Selecting the "Uninitialized" option resets the parameters to their default values and then restarts the execution of the control strategy. The "Processor" performs a CPU reset. During normal operation this parameter displays Run.

The coordination of the function block execution is configured by the cycle select parameter (CYCLE_SEL) from the available options identified by the cycle type parameter (CYCLE_TYPE). There are three possible options:

- Scheduled
- Event driven
- Manufacturer specific

The minimum cycle time parameter (MIN_CYCLE_T) indicates the shortest macro cycle the device can accept. This parameter is set by the manufacturer in units of 1/32 of a millisecond, just like the report confirmation time. The cycle time parameters are explained in more detail in the section on FOUNDATION function blocks.

The special characteristics of the device are listed in the features (FEATURES) parameter. This parameter cannot be changed. The device features that are covered are the following:

- Unicode strings
- Reports

- Fault state
- Soft write lock
- Hard write lock
- Output readback
- Direct output write
- Change of bypass in Auto allowed
- MVC supported

Using the feature select (FEATURE_SEL) parameter it is possible to choose which optional features of the device can be used. For example, the software write locking can be activated to prevent configuration change. The feature options are best described in the context of their respective features. "Change of bypass in Auto allowed" is discussed in the section on PID block, "Output readback" and "Fault state" are covered in the section on the output class of blocks, and "Reports" is discussed in the section on alarms. If the configuration tool has configured strings using Unicode characters instead of ASCII—for example, for the tag descriptor—this is indicated by the feature select parameter. Multi-Variable Container (MVC) is a feature used to reduce communication overhead and thereby improve communication efficiency. If both the device and host support MVC, it is a good idea to enable it.

If the device has a write lock to protect the configuration from unauthorized tampering or accidental change the operator can, in the event of soft write lock, set the write lock (WRITE_LOCK) parameter, to protect the configuration from external writes. In the event of hardware write lock the parameter can only be read. In the event of soft write lock the protection may be switched off by clearing the write lock, which will generate an alert to the host. The priority of the write alarm is configured by the write priority parameter (WRITE_PRI) in the same way as for the other alarms described in the section on FOUNDATION function block. The write alarm parameter (WRITE_ALM) contains the information for the latest occurrence of the write alarm. Remember that you have to disable the write lock parameter before the software write lock in the feature select parameter can be deactivated.

Most configuration tools internally keep track of the resources in each device, the resources required for each function, and how much of the resources are available. It is also possible to check this information on line in the device. The free memory and execution time parameters (FREE_SPACE) and (FREE_TIME) inform you of the percentage of available memory and execution time, respectively. The available memory in kilobytes can be read from the memory size (MEMORY_SIZE) parameter.

If the device supports Fault State for the output class blocks, all the output class function blocks in the device may immediately be forced into the fail-safe option that they have been configured for, thus shutting the loops down. Fault State is forced by the set fail-safe (SET_FSTATE) parameter, which ignores the fault state time-out setting in the output block. The Fault State mode may be cleared by the clear fail-safe parameter (CLR_FSTATE). The current state of the force function may be read from the fail-safe parameter (FAIL_STATE).

There is a time limit before mode shedding occurs in the function blocks as a result of communication loss for all the remote cascade parameters in the device. This limit is configured by the remote cascade shed time-out parameter (SHED_RCAS), and for the remote output parameters it is configuration by the remote output shed time-out (SHED_ROUT). These parameters are configured in units of 1/32 of a millisecond, just like the report confirmation time. This is explained in greater detail for the RCas and ROut modes in the section on function blocks.

The GRANT_DENY, UPDATE_EVT, BLOCK_ALM, ALARM_SUM, and ACK_OPTION parameters are described in the section on FOUNDATION function blocks.

Humans do not utilize the read and write test parameter (TEST_RW); it is provided solely for device testing.

Minimum Configuration

At a minimum, the following parameters must be set:

MODE_BLK.Target

Transducer Blocks

Transducer blocks interface the function blocks to the device I/O hardware such as sensors, actuators, and display (figure 4-11). These blocks are responsible for the information and functionality specific to, for example, the measurement of a particular physical property, such as pressure or temperature. Note that an analog input block is the same in a pressure as in a temperature transmitter. The transducer blocks have parameters for handling the features that set one type of transmitter apart from another. Transducer blocks handle not only measurement but also actuation and display. For this purpose, there are three types of transducer blocks:

- Input transducer (transmitters and analyzers)
- Output transducer (final control elements)
- Display transducer

All the parameters in the transducer blocks are contained and cannot therefore be linked like function blocks. The input and output transducer blocks are associated with the input and output class function blocks, respectively. The transducer blocks interface to the function blocks over I/O hardware channels, which are different from function block links. The function blocks are assigned to their transducer block through an I/O hardware channel (CHANNEL) parameter in the corresponding input or output function block. Function blocks can only be assigned to transducer blocks in the same device.

For a majority of devices, engineers will need to configure very few parameters in the transducer block because most of them only indicate limits and diagnostics. In most cases, only the mode parameter has to be set, and sometimes maybe one or two additional parameters. The transducer blocks for devices made by different manufacturers are different, although the first two dozen or so parameters often are the same. In other words, fieldbus devices from different manufacturers are not identical. A transducer block configured for one manufacturer's device can therefore not be used in another manufacturer's device. However, since only one or two parameters have to be set to configure the device it is still very simple to replace one fieldbus device with that of another manufacturer.

In addition to the universal parameters all transducer blocks have a few standard parameters. The block error (BLOCK_ERR) parameter indicates the classification of any failures that were detected by the device's diagnostics. The transducer error (XD_ERROR) parameter indicates the diagnostics with a greater level of detail. The errors that the self-diagnostics check for are manufacturer specific. Only the highest-priority transducer error is indicated. The transducer error parameter is discussed in greater detail in chapter 6, "Troubleshooting." The UPDATE_EVT and BLOCK_ALM parameters are described in the section on FOUNDATION function blocks.

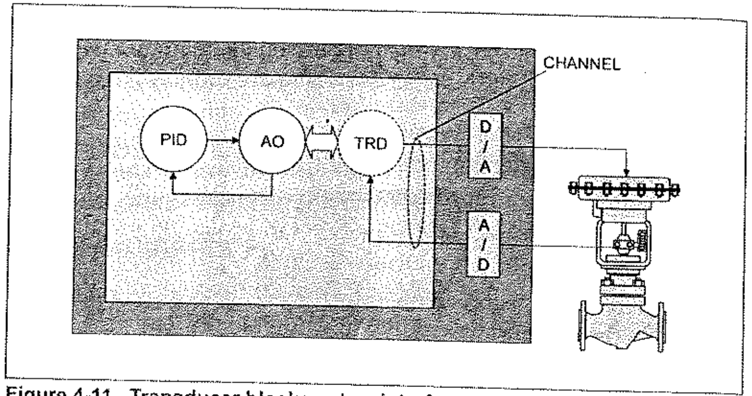


Figure 4-11. Transducer blocks act as interfaces to hardware.

Typically, a device has one transducer block for each measurement or actuation the device performs. The transducer blocks for each measurement or actuation that a multiple variable device performs may not be available. In this case, a configuration tool may use the parameters TRANSDUCER_DIRECTORY, TRANSDUCER_TYPE, and COLLECTION_DIRECTORY to locate information for individual I/Os inside a single transducer block. These parameters are not really useful for humans.

Common Analog Input

Input transducer blocks are found in transmitters and analyzers, that is, devices attached to sensors. In addition to the standard transducer block parameters the primary value type (PRIMARY_VALUE_TYPE) parameter indicates what type of measurement the device performs and is therefore useful for informational purposes. The primary value (PRIMARY_VALUE) parameter comes from the sensor and goes to the AI block over the hardware channel. It shows the measured value in the engineering unit set in the primary value range (PRIMARY_VALUE_RANGE) parameter. The primary value range parameter is usually set from the AI block transducer scale (XD_SCALE) parameter and mirrors those values to prevent any inconsistencies and subsequent errors. The primary value range parameter has no function in the transducer block other than as information for the user.

The type of sensor with which the transmitter is fitted is identified by the sensor type parameter (SENSOR_TYPE). For some device types, such as temperature, this parameter can be set by the user. The serial number of the sensor or sensor module of the transmitter is shown by the sensor serial number (SENSOR_SN) parameter, which can be up to thirty-two characters long. The operating limits for the sensor type may be read from the high- and low-range elements in the sensor range limit (SENSOR_RANGE) parameter but cannot be changed since they are capabilities determined by the physical characteristics of the device. Just as with scaling parameters, the sensor range parameter has four elements: the lower and upper sensing limits, the engineering unit the values are displayed with, and the number of decimals the value should be displayed with. The block diagnostics checks for configuration errors and reports a block alarm for any inconsistencies it finds, for example, if the AI block transducer scale (XD_SCALE) is set beyond the sensor's range limits.

Calibration (sometimes referred to as sensor trim) should not be confused with range setting. If range is set at all it is usually done in the PID function block or in the transducer scale (XD_SCALE) parameter of the AI block to prevent any inconsistencies and subsequent errors. Sensor calibration is performed by writing the low and high calibration point parameters (CAL_POINT_LO and CAL_POINT_HI, respectively) with the value of the applied input (figure 4-12). For example, to calibrate the zero of a pressure transmitter the input may be equalized and vented to ensure that zero pressure is applied. Once the reading seen at the primary value, which presumably is a bit incorrect due to drift, has stabilized, the actual applied input, in this example 0, is written as the low calibration point parameter. Similarly, the span may be calibrated by applying pressure from a precision source or by comparing it against a standard. Calibration is discussed in greater detail in chapter 9, "Maintenance and Asset Management."

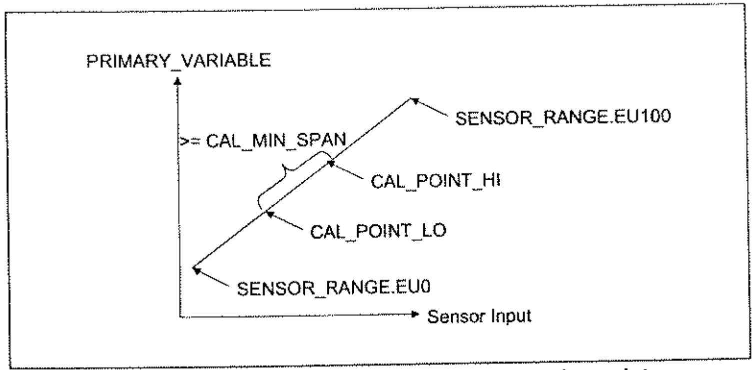


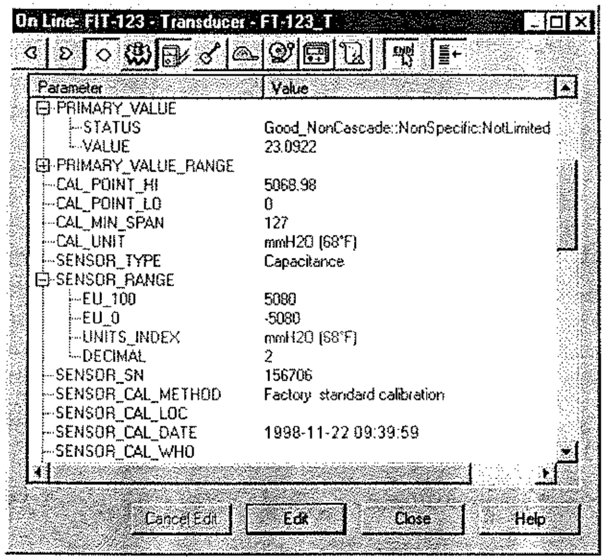
Figure 4-12. Relation between sensor range and calibration points.

Since the calibration values are written and stored in the two calibration point parameters they also indicate at which values calibration was last performed. This makes it is easy to check if the transmitter was last calibrated for a range suitable for the application it is operating in. The smallest span—that is, the smallest absolute difference between the calibration point values—for which the sensor may be calibrated is informed by the calibration minimum span value (CAL_MIN_SPAN). The unit of the calibration point values is chosen with the calibration unit parameter (CAL_UNIT). Changing the calibration unit automatically updates all the associated values to the new unit. The calibration unit comes in handy if you want to calibrate the transmitter in a unit other than the one in which indication is normally done. For example, the AI block XD_SCALE in a pressure transmitter may be set in kPa, but the deadweight tester weights come in mbar.

To leave a calibration trail, the technician may store and verify the data about the latest calibration in the device, including the date, the location, the method used, and who calibrated it (figure 4-13). The time and date is stored in the sensor calibration date (SENSOR_CAL_DATE) parameter. The sensor calibration location (SENSOR_CAL_LOC) parameter can be up to thirty-two characters long, for example, "in-situ," "workshop," or "NIST," and informs the location where the last calibration was done. The name of the person who carried out the calibration is retained and checked from the "sensor calibrate whom" (SENSOR_CAL_WHO) parameter, which can be up to thirty-two characters in length. The procedure used for the calibration may be recorded in the sensor calibration method (SENSOR_CAL_METHOD) parameter, which has the following options:

- Volumetric
- Static weigh
- Dynamic weigh
- Factory standard calibration
- User standard calibration
- Factory special calibration
- User special calibration
- Other

It is a good idea to update the calibration record for the transducer block each time you have done a calibration in order to simplify calibration audits.



Parameter	Value
PRIMARY_VALUE	
STATUS	Good_NonCascade::NonSpecific:NotLimited
VALUE	23.0922
PRIMARY_VALUE_RANGE	
CAL_POINT_HI	5068.98
CAL_POINT_LO	0
CAL_MIN_SPAN	127
CAL_UNIT	mmH20 (68°F)
SENSOR_TYPE	Capacitance
SENSOR_RANGE	
EU_100	5080
EU_0	-5080
UNITS_INDEX	mmH20 (68°F)
DECIMAL	2
SENSOR_SN	156706
SENSOR_CAL_METHOD	Factory standard calibration
SENSOR_CAL_LOC	
SENSOR_CAL_DATE	1998-11-22 09:39:59
SENSOR_CAL_WHO	

Figure 4-13. Sensor range and calibration information.

Most measurements require that an auxiliary sensor measure a secondary value in order to compensate for such influences on the primary value as ambient temperature, and the like. This second sensor measurement can be seen from the secondary value (SECONDARY_VALUE) parameter, which is displayed in the engineering unit set in the secondary value unit (SECONDARY_VALUE_UNIT) parameter. For most transmitter types the secondary value is temperature.

Common Analog Output

Output transducer blocks are found in devices like valve positioners and current as well as in pneumatic output converters connected to final control elements. In addition to the standard transducer parameters, the common output block parameters are almost all the parameters that a simple output converter needs. A valve positioner with a servo-positioning algorithm has additional standard parameters. The final value (FINAL_VALUE) parameter comes from the AO block with the manipulated variable for the final control element over the hardware channel. It shows the desired output value in the engineering unit that was set in the final value range (FINAL_VALUE_RANGE) parameter. The final value range also indicates the desired output operating range for basic converters. It is usually set from the transducer scale (XD_SCALE) parameter in the AO function block to prevent any inconsistencies and subsequent errors. For example, it may have been set to 3-15 psi or 4-20 mA in a signal converter, but it is usually 0-100% in a valve positioner.

Depending on the characteristics of the positioner, actuator, or valve package, the actuator fail action (ACT_FAIL_ACTION) parameter may or may not affect the operation of the output transducer block and the actuator. Most actuators have, and should have, a mechanical fail-safe position that takes effect when all else fails, and the actuator fail action merely reflects this characteristic for informational purposes. In either case, it is a good idea to configure the parameter correctly if only to provide as information for asset management. The options for the actuator fail action parameter are as follows:

- Self-closing
- Self-opening
- Hold last value
- Maximum value
- Minimum value
- Indeterminate

Configure the valve type (VALVE_TYPE) parameter according to the type of valve/actuator the positioner is mounted on. The options for the valve type parameter are the following:

- Linear
- Rotary
- Other

You may store identification information for the control valve for asset management purposes. The valve identification consists of a manufacturer parameter (VALVE_MAN_ID), a thirty-two-character model parameter (VALVE_MODEL_NUM), and a thirty-two-character serial number parameter (VALVE_SN). The actuator identification parameters ACT_MAN_ID, ACT_MODEL_NUM, and ACT_SN have the same function and characteristics as the valve identification parameters VALVE_MAN_ID, VALVE_MODEL_NUM, and VALVE_SN.

The calibration trail parameters XD_CAL_LOC, XD_CAL_DATE, and XD_CAL_WHO have the same function and characteristics as SENSOR_CAL_LOC, SENSOR_CAL_DATE, and SENSOR_CAL_WHO in the input transducer block, respectively. Valve positioners usually have an automatic setup feature that calibrates the travel to the fully opened and closed positions of the valve.

Pressure Transducer

Pressure transducer blocks are found in pressure transmitters. The primary value type (PRIMARY_VALUE_TYPE) parameter indicates whether the sensor is for differential pressure, gauge pressure, or absolute pressure. The sensor type (SENSOR_TYPE) parameter indicates which technology is used in the pressure sensor: capacitance, resonant mode strain gauge, resistive mode strain gauge, and so on.

The primary value (PRIMARY_VALUE) parameter is always the measured pressure that is passed to the AI function block over the hardware channel. The AI block may convert the pressure reading into flow, level, or other inferred measurement. The secondary value is usually the sensor module temperature, which is useful for checking whether the transmitter is within operating limits. In addition to the standard input transducer parameter, you can use the sensor isolator material (SENSOR_ISOLATOR_MTL) parameter to see the material of construction for the sensor diaphragm. Likewise, the fill fluid of the sensor the may be seen from the sensor fill fluid (SENSOR_FILL_FLUID) parameter.

Temperature Transducer

Temperature transducer blocks are found in temperature transmitters. The type and characteristics standard for the temperature sensor is configured through the sensor type (SENSOR_TYPE) parameter. The primary value (PRIMARY_VALUE) parameter is the measured temperature, or in some cases optionally a pure voltage or resistance, which is passed to the AI function block over the hardware channel. In addition to the standard input transducer parameter, you can use the sensor connection (SENSOR_CONNECTION) parameter to configure the electrical connection of the sensor to perform the appropriate lead-wire compensation, and the like. The standard sensor connection options are as follows:

- Two-wire
- Three-wire
- Four-wire

The secondary value (SECONDARY_VALUE) parameter is the cold-junction temperature that is used for thermocouple compensation. It may be useful to know when you are checking the calibration of thermocouples. Some temperature transmitters are constructed as an individual-module "hockey puck" that goes into the transmitter housing. The module serial number (MODULE_SN) may be used to track the device.

Valve Positioner Transducer (Advanced)

Advanced valve positioner transducer blocks are found in control valve positioners that have a servo-positioning PID algorithm for controlling the valve. In addition to the basic positioning analog output transducer parameters, the advanced positioner also has parameters for feedback and servo tuning. The final value range (FINAL_VALUE_RANGE) in a positioner would typically be set as 0-100% from the transducer scale (XD_SCALE) parameter in the AO function block. The desired valve position is received from the AO block over the hardware channel at the final value (FINAL_VALUE) parameter. To provide tight shutoff, you can set low and high cutoff limits for the final value in the final value cutoff low and high (FINAL_VALUE_CUTOFF_LO and FINAL_VALUE_CUTOFF_HI) parameters, respectively. If the final value is below the low cutoff value the servo will shut the valve tight. Likewise, if the final value exceeds the high cutoff value the servo will open the valve wide (figure 4-14).

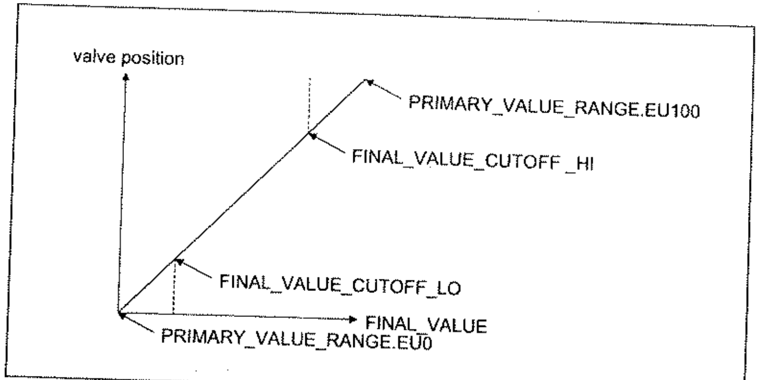


Figure 4-14. Tight shutoff.

The actual valve position value as sensed by the positioner feedback sensor is indicated in the final position value (FINAL_POSITION_VALUE). This value is also fed back to the AO function block over the hardware channel, and there it is displayed as the readback (READBACK) parameter. As explained in detail with respect to the AO function block, you may optionally use the actual valve position to initialize the PID. This will provide true bumpless transfer when an open loop is once again closed.

You can tune the response of the servo-positioning algorithm to suit the characteristics of small or large valves. The PID gains for the servo are configured in the servo gain (proportional), servo reset (integral), and servo rate (derivative) parameters (SERVO_GAIN, SERVO_RESET and SERVO_RATE), respectively.

PROFIBUS PA Devices

The process of configuring devices consists of entering the selection of the devices that you will used as well as the configuration and parameterization of the blocks. In every PA device, you must configure a physical block similar to the FOUNDATION resource block. Physical device tags are not used. Sensors and actuators are configured in their corresponding transducer blocks.

PA function block links between devices will be handled in the future. PA function blocks are therefore not linked into control strategy. Rather, they are the point where the central controllers cyclically read and writes process variables and desired valve positions. Therefore, the parameterization of the PA input and output block is a part of device configuration along with the physical and transducer blocks.

Blocks

As in FOUNDATION Fieldbus, the PROFIBUS PA blocks are logical groupings of parameters. For the PA blocks, you should configure the tag name in the TAG_DESC parameter. It must be unique in the system and can be up to thirty-two characters in length, for example, "PIT-2308."

The PA profile includes transducer blocks for several different device types. Each block has a set of parameters for the common functions of the device type. Manufacturers also add an extended set of parameters to cater to the specifics of the sensing or actuation technology employed. Standard device description files will be added later. Therefore, host and device manufacturers have to work together to develop device support files in order to make interoperation in open systems possible.

Parameters and Status

PROFIBUS PA parameters have characteristics like those of FOUNDATION when it comes to reading, writing, storage, and so on. The concept of status is also the same.

Universal Parameters

Every PA block has seven universal parameters. Function blocks have one additional universal parameter. The ST_REV, STRATEGY, and ALERT_KEY parameters function in the same way as they do for FOUNDATION although events and alarms will only be handled in the future. The TAG_DESC parameter is used as the tag name to identify the block rather than as a function descriptor. Therefore, the TAG_DESC parameter must be parameterized so it is unique within the system. MODE_BLK is similar to the FOUNDATION parameter only that the Target mode, instead of being an element in the MODE_BLK parameter, is a parameter on its own, namely, the target mode (TARGET_MODE) parameter. The PA physical block and the transducer blocks generally do not use the block mode parameter, so there is one thing less that has to be configured. PA function blocks may support the OOS (out-of-service), Man (Manual), Auto (Automatic), and RCas (Remote Cascade) modes. Advanced, class B, devices may also support the LO (Local Override) mode. In PA, unlike FOUNDATION, the ALARM_SUM parameter is also universal. Though it works in a very similar way to the parameter in FOUNDATION, it currently only supports the indication of the Current states of the alarms to show which ones are active.

The batch (BATCH) parameter does not affect the operation of the function block, but it may be used for identification purposes in batch control applications. The batch parameter has four elements, based on the ANSI/ISA-88.01-1995 standard: batch identification (BATCH_ID) is a number from 0 to 4,294,967,395; the recipe unit procedure (RUP) is a number from 0 to 65,535; the recipe operation (OPERATION) is a number from 0 to 65,535; and the recipe phase (PHASE) is a number from 0 to 65,535. See the separate section on introduction to the ANSI/ISA-S88.01-1995 standard.

Physical Block

Though the PA physical block is similar to the FOUNDATION resource block, the block mode parameter is typically not used.

There are a number of parameters for identifying the device. The manufacturer of the device is identified by the device manufacturer identification (DEVICE_MAN_ID) parameter. The device type identification (DEVICE_ID) parameter identifies the basic model of the device, and the revision of the device software and hardware is indicated by the software revision (SOFTWARE_REVISION) and hardware revision (HARDWARE_REVISION) parameters, respectively. The device is also identified by its serial number in the device serial number (DEVICE_SER_NUM) parameter. The ident number control (IDENT_NUMBER) is used to select either the profile-specific or manufacturer-specific ident number. There are different profiles for the various combinations of resource, transducer, and function blocks that a device can have. The ident number tells the configuration tool which one of these profiles the device belongs to, that is, how the host will interact with the device. It is not really useful for humans.

A summary of the current hardware and software error status in the device is kept in the diagnostics (DIAGNOSIS) parameter. The checks it runs include:

- Electronic failure
- Mechanics failure
- Motor temperature too high
- Electronic temperature too high
- Memory error
- Failure in measurement
- Device not initialized (No self-calibration)
- Self-calibration failed
- Zero point error (limit position)
- Power supply failed (electrical, pneumatic)
- Configuration not valid
- New-start-up (warm start-up) carried out.
- Re-start-up (cold start-up) carried out.
- Maintenance required
- Characterization invalid

Additional self-diagnostic checks continue in the manufacturer-specific diagnostics extension (DIAGNOSIS_EXTENSION) parameter. A configuration tool may use the parameters

DIAGNOSIS_MASK and DIAGNOSIS_MASK_EXTENSION to filter out the checks that are not performed by the device. These two parameters are not really useful for humans.

If the device has a hardware write lock switch the hardware write-protection (HW_WRITE_PROTECTTION) parameter can be read to check whether the device is write locked or unprotected. Software write-protection lock can be set (protected) or cleared (unprotected) using the write lock (WRITE_LOCKING) parameter. Local adjustment switches can be enabled or disabled using the local operation enable (LOCAL_OP_ENA) parameter. If a communication failure lasts for longer than half a minute the device will enable the local adjustment; this will make it possible to troubleshoot and perform maintenance.

Through the factory reset (FACTORY_RESET) parameter there are three ways to reset the device:

- Default values, maintain address
- Plain restart
- Address reset

Like HART, the PA devices provide for basic device identification. The thirty-two-character descriptor (DESCRIPTOR) parameter may be used to describe its application, for example, "Boiler 1 drum level." Special considerations or instructions can be stored in a thirty-two-character message (DEVICE_MESSAGE) parameter, for example, "ladder needed for access." Lastly, the date of installation can also be entered in the date (DEVICE_INSTAL_DATE) parameter. The read-only device certification (DEVICE_CERTIFICATION) parameter can hold up to thirty-two characters of manufacturer-entered data regarding approvals for the device.

Transducer Blocks

The block mode parameter is typically not used in the PROFIBUS PA transducer blocks. Most of the parameters in the transducer blocks are unique, but all analog input transducer blocks have a primary value (PRIMARY_VALUE) parameter, which is the measured value. For flexibility in linearization the operator can enter a lookup table. There are several parameters that the host can use to display the table; all have a name that starts with TAB_. The number-of-points (TAB_MAX_NUMBER) parameter indicates the maximum number of points that the table can have.

Pressure Transducer

The sensor value (SENSOR_VALUE) parameter contains the noncalibrated reading from the sensor in a unit that is selected by the sensor unit (SENSOR_UNIT) parameter, which is also used for calibration. The calibration is done in a way that is similar to the way FOUNDATION transducer blocks obtaining the calibrated value (TRIMMED_VALUE). The sensor type (SENSOR_TYPE) parameter works just like as it does the FOUNDATION transducer.

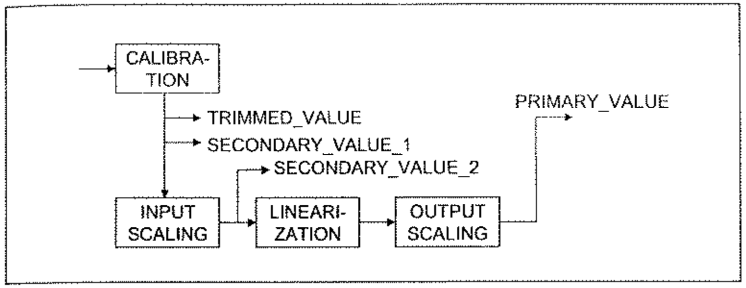


Figure 4-15. PROFIBUS PA pressure transducer schematic diagram.

Linearization

In addition to converting pressure into inferred measurements for flow, level, and the like in the AI block, for PROFIBUS PA this same conversion can also be done in the transducer block. It is important to make sure that the square root is not extracted in both the transducer and the AI blocks. It may be a good idea to adopt a plantwide strategy for consistent implementation. To avoid any confusion, always do square root extraction in the AI block. The primary value type (PRIMARY_VALUE_TYPE) parameter selects the type of measurement that is done for the application. That is, the conversion for the primary value (PRIMARY_VALUE) parameter passed to the AI block and the options available for the unit (PRIMARY_VALUE_UNIT). The first secondary value (SECONDARY_VALUE_1) is identical to the pressure reading. The second secondary value (SECONDARY_VALUE_2) is the calibrated pressure reading, which is scaled to percentage over the range set in the input scaling (SCALE_IN) parameter. After input scaling, the linearization determined by the primary type is applied to the second secondary value. This is followed by output scaling back to an engineering unit to obtain the primary value (figure 4-15). The available primary value types are as follows:

- Pressure

- Flow
- Level
- Volume

When "Pressure" is selected, the pressure reading becomes the primary value without further linearization. The other options are used for inferred measurements, and the input scale and output scale parameters come into play. When "Flow" is selected, the square root is extracted from the second secondary value. The engineering unit and range of the actual pressure measurement is set in the input scale, and the engineering unit and range of the inferred flow measurement is set in the output scale. When "Level" is selected, there is a linear conversion from the second secondary value. In other words, the function is very similar to that described for the FOUNDATION AI block. The LOW_FLOW_CUT_OFF parameter has the same function as the LOW_CUT FOUNDATION parameter but is set in percentage. The linear point (FLOW_LIN_SQRT_POINT) parameter has a similar function, but rather than simply setting the reading to zero it is linearly interpolated from zero to this point (figure 4-16).

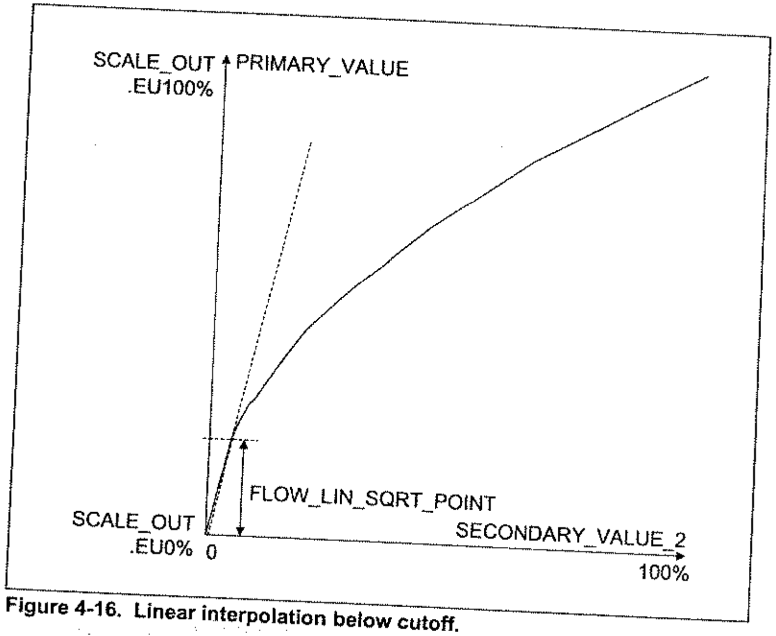


Figure 4-16. Linear interpolation below cutoff.

The units associated with the two secondary values are set in the SECONDARY_VALUE_1_UNIT and SECONDARY_VALUE_2_UNIT parameters, respectively.

Maintenance

The operating limits for the sensor type may be read from the high- and low-sensor range limit (SENSOR_HI_LIM and SENSOR_LO_LIM) parameters. However, they cannot be changed since they are capabilities that are determined by physical characteristics of the device. The sensor temperature is displayed in the temperature (TEMPERATURE) parameter in a unit that is selected by the temperature unit (TEMPERATURE_UNIT) parameter. Other information for facilitating asset management is also stored in the transducer block: sensor serial number (SENSOR_SERIAL_NUMBER), maximum static pressure for the sensor (SENSOR_MAX_STATIC_PRESSURE), the type of process connection (PROCESS_CONNECTION_TYPE), and the material of construction (PROCESS_CONNECTION_MATERIAL). Other asset-management information includes sensor fill-fluid (SENSOR_FILL_FLUID), material of construction for the sensor diaphragm (SENSOR_DIAPHRAGM_MATERIAL), and o-ring (SENSOR_O_RING_MATERIAL).

The extreme conditions for pressure and temperature are captured in four parameters: MAX_SENSOR_VALUE, MIN_SENSOR_VALUE, MAX_TEMPERATURE, and MIN_TEMPERATURE. These may be used to determine if a sensor fault could possibly be caused by exposure to conditions that are beyond operational limits.

Temperature Transducer

The temperature transducer handles up to two inputs. There are therefore two sets of parameters, one for each measurement point. The two inputs must have the same sensor type and display in the same unit. Additionally, the inputs have a few parameters in common. The parameters for the individual inputs have names ending with _1 and _2, respectively. For example, after the addition of a bias (BIAS_*) parameter the two temperature readings are displayed as SECONDARY_VALUE_1 and SECONDARY_VALUE_2, respectively.

The measurement type (SENSOR_MEASUREMENT_TYPE) parameter selects the measurement that is displayed in the primary value (PRIMARY_VALUE) parameter passed to the AI block. The available options for obtaining the primary value are as follows:

- Input 1
- Input 2
- Input 1 - input 2
- Input 2 - input 1
- Average of input 1 and input 2
- Average of input 1 and input 2, or "Good" if one goes "Bad"

The unit for the primary value, the two secondary values, and the bias parameters is set in the primary value unit (PRIMARY_VALUE_UNIT) parameter. The linearization type (LIN_TYPE) parameter selects the type of temperature sensor that is used for both inputs. The available options depend on the device but typically include all standard thermocouples and RTDs.

Maintenance

The operating limits for the sensor type may be read from the high- and low-range sensor limit (UPPER_SENSOR_LIMIT and LOWER_SENSOR_LIMIT) parameters. However, they cannot be changed since they are capabilities that are determined by the physical characteristics of the device and sensor.

A summary of the current status for the malfunctions that affect both inputs is kept in the general input fault (INPUT_FAULT_GEN) parameter. Standard checks include:

- Reference junction error
- Hardware error

A summary of the current status for each individual input is kept in the input fault (INPUT_FAULT_*) parameters. Standard checks include:

- Under range
- Over range
- Lead breakage
- Short circuit

The individual input diagnostics are performed, provided they are enabled from the sensor wire check (SENSOR_WIRE_CHECK_*) parameters. A total of four combinations of lead breakage and short circuit are possible.

The extreme conditions for the two inputs are captured in the parameters MAX_SENSOR_VALUE_* and MIN_SENSOR_VALUE_*. They are reset by writing to them.

Thermocouple

In addition to the standard temperature transducer parameter the cold-junction temperature (RJ_TEMP) shows the temperature measured at the sensor terminals. The reference junction type (RJ_TYPE) parameter selects the type of cold-junction compensation that will be performed. The options are the following:

- None
- Internal
- External

If the "None" option is selected no cold-junction compensation will be performed. If you select the "Internal" option the cold-junction temperature measured at the sensor terminals will be used. If you select the "External" option the value manually entered at the external reference junction temperature (EXTERNAL_RJ_VALUE) parameter will be used for the compensation.

RTD

In addition to the standard temperature transducer parameter, the sensor connection (SENSOR_CONNECTION) parameter is used to configure the electrical connection of the sensor that is to perform the appropriate lead-wire compensation and so on. The standard sensor connection options are as follows:

- Two-wire
- Three-wire
- Four-wire

For a two-wire connection, you can manually enter the lead-wire resistance, in ohms, in the wire compensation (COMP_WIRE*) parameters.

Electropneumatic Positioner Transducer

The demanded position value (POSITIONING_VALUE) parameter with the manipulated variable comes from the AO block for the final control element over the hardware channel. It shows the desired output value in units that are set in the output scale (OUT_SCALE) parameter in the AO function block. To provide tight shutoff, the engineer can set the low and high cutoff limits for the final value in the final value cutoff low and high (SETP_CUTOFF_DEC and SETP_CUTOFF_INC) parameters, respectively. These are similar to the FOUNDATION parameters FINAL_VALUE_CUTOFF_LO and FINAL_VALUE_CUTOFF_HI, respectively.

The engineer can tune the response of the servo-positioning algorithm to suit the characteristics of small or large valves. The PID gains for the servo are configured in the servo gain (proportional), servo reset (integral) and servo rate (derivative) parameters, which are similar to the FOUNDATION parameters. PROFIBUS PA provides two sets of gains, one for increasing and the other for decreasing. They are denoted by names ending with _1 and _2, respectively: (SERVO_GAIN_*, SERVO_RESET_* and SERVO_RATE_*).

The actuator fail-safe position (ACTUATOR_ACTION) parameter is similar to the FOUNDATION ACT_FAIL_ACTION parameter. The options for the actuator fail action parameter are as follows:

- Self-closing
- Self-opening
- None/Hold last value

The valve type (VALVE_TYPE) parameter is similar to the FOUNDATION. The options for the valve type parameter are these:

- Linear
- Rotary - part turn
- Rotary - multi turn

The actual valve position feedback value (FEEDBACK_VALUE) is similar to the FOUNDATION FINAL_POSITION_VALUE parameter. Deviations between the demanded position and the actual position that are smaller than the deadband (DEADBAND) are not acted upon. Use this feature to reduce air consumption and valve wear. This parameter value is set in percentage terms to balance the interests of control stability and product variability.

TRAVEL_LIMIT_LOW

TRAVEL_LIMIT_UP

RATED_TRAVEL

There are parameters for increasing (TRAVEL_RATE_INC) and decreasing (TRAVEL_RATE_DEC) rate-of-change, which operate like the FOUNDATION function block parameters SP_RATE_UP and SP_RATE_DN, respectively.

Linearization

Linearization is set by the linearization type (LIN_TYPE) parameter and can be used to achieve applied flow characterization (figure 4-17). The available linearization types are the following:

- None (linear)
- Equal percentage 1:33 rangeability
- Quick opening 1:33 rangeability
- Equal percentage 1:50 rangeability
- Quick opening 1:50 rangeability
- Equal percentage 1:25 rangeability
- Quick opening 1:25 rangeability
- Table

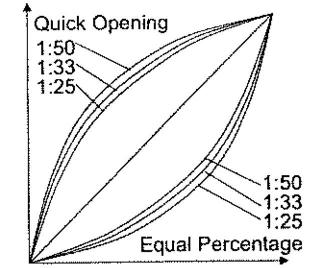


Figure 4-17. Applied flow characteristics.

Maintenance

Identification information for the control valve may be stored for asset management purposes. The valve identification consists of a sixteen-character manufacturer parameter (VALVE_MAN) and a sixteen-character serial number parameter (VALVE_SER_NUM). The actuator identification parameters ACTUATOR_MAN and ACTUATOR_SER_NUM have the same function and characteristics as the valve identification parameters VALVE_MAN and VALVE_SER_NUM. If an air booster is added its asset information can be stored too. The booster identification consists of a sixteen-character manufacturer parameter (ADD_GEAR_MAN), a sixteen-character serial number parameter (ADD_GEAR_SER_NUM), and a sixteen-character type (ADD_GEAR_ID). The type of actuator is displayed in the actuator type (ACTUATOR_TYPE) parameter. The possible types are these:

- Electro-pneumatic
- Electric
- Electro-hydraulic
- Others

The opening and closing stroking times for the valve package at the time of commissioning are stored in the ACT_STROKE_TIME_INC and ACT_STROKE_TIME_DEC parameters, respectively.

The dates on which several important functions were carried out are stored in sixteen-character fields. The dates stored are the last valve maintenance (VALVE_MAINT_DATE), last booster installation (ADD_GEAR_INST_DATE), last calibration (DEVICE_CALIB_DATE), and last configuration (DEVICE_CONFIG_DATE).

The total travel is accumulated and displayed in the equivalent number of full strokes in the total travel (TOTAL_VALVE_TRAVEL) parameter. In other words, it is like an odometer that may be used to target proactive maintenance by setting the total travel limit (TOT_VALVE_TRAV_LIM) parameter to a value at which the maintenance should be planned.

Auto-calibration of travel is invoked using the self-calibration (SELF_CALIB_CMD) parameter. Although the procedure is device-specific, the means for invoking it is standardized. The available options are these:

- Normal operation
- Zero calibration

- Self-calibration
- Reset the accumulated travel (odometer)

The current status for the progress of the auto-calibration can be seen from the self-calibration status (SELF_CALIB_STATUS) parameter. The possible states are as follows:

- Aborted
- Mechanical error
- Time-out
- Emergency override
- Zero error
- No valid data

Function Blocks

The PROFIBUS PA input and output function blocks are very similar to the FOUNDATION Fieldbus counterparts. The function blocks are described in the section of this chapter on the FOUNDATION Fieldbus programming language, so we will point out only the differences here.

Controller Link

The main difference is that PROFIBUS function blocks cannot be linked to each other to build a control strategy. In other words, to form a control loop the value from an AI block is read by a central controller that is configured in whatever programming language, and the manipulated variable is written to the AO block. Because function block links are not used, the RCas mode is the most important mode for output blocks. The manipulated variable is written to the remote cascade setpoint input (RCAS_IN), and for the cascade to be established the controller must handshake with the remote cascade setpoint output (RCAS_OUT) parameter of the output block. The remote cascade setpoint input parameter and the remote cascade setpoint output are input and output parameters, respectively, rather than contained within the block. This handshake is very similar to the initialization that is performed in the FOUNDATION Fieldbus AO block using the CAS_IN and BKCAL_OUT parameters.

It may be a good idea in a system using PROFIBUS PA devices to use a controller that has the ability to fully utilize the status for the parameter values in both input and output function blocks. The controller should *have the option of initiating fail-safe if it receives a "Bad" input reading, for example, a sensor failure or communication error. Likewise, the controller should utilize the output block limit status to prevent reset windup and to provide bumpless mode transfers.*

Alarm Parameters

The PROFIBUS PA alarm parameters are very similar to the Foundation Fieldbus parameters. However, PROFIBUS PA does not support the alert-reporting mechanism, and a host must therefore poll the alarm status parameters to detect the alarm and provide the time stamp.

AI - Analog Input Function Block

The PROFIBUS PA analog input block is very similar to the Foundation Fieldbus AI block. In PROFIBUS PA, the XD_SCALE parameter is called PV_SCALE, and the L_TYPE parameter is called LIN_TYPE.

When the primary value from the transducer block is "Bad," the fail-safe type (FSAFE_ TYPE) parameter configures three options regarding what should happen:

- Fail-safe
- Last Usable Value
- Bad

For the "Fail-safe" option in the event of input failure the output (OUT) will be set to the value configured in the fail-safe value (FSAFE_VALUE) parameter with a status of "Uncertain." For the "Last Usable Value" option, the last valid value becomes the output with a status of "Uncertain." For the "Bad" option, the status becomes "Bad."

In addition to the predefined standard units, the user can also choose a "textual unit definition" and a sixteen-character label that are entered in the unit text (OUT_UNIT_TEXT) parameter.

AO - Analog Output Function Block

The PROFIBUS PA analog output block is similar to the Foundation Fieldbus AO block. The main difference is that in PROFIBUS PA, the manipulated variable from the controller is received on the remote cascade setpoint input (RCAS_IN), typically in percentage terms. The main operating mode for the block is RCas. In PROFIBUS PA, the XD_SCALE parameter is called OUT_SCALE. Rather than just a single CHANNEL parameter, there are two separate parameters: OUT_CHANNEL, which identifies the transducer block that the desired output value goes to, and IN_CHANNEL, which identifies the transducer block

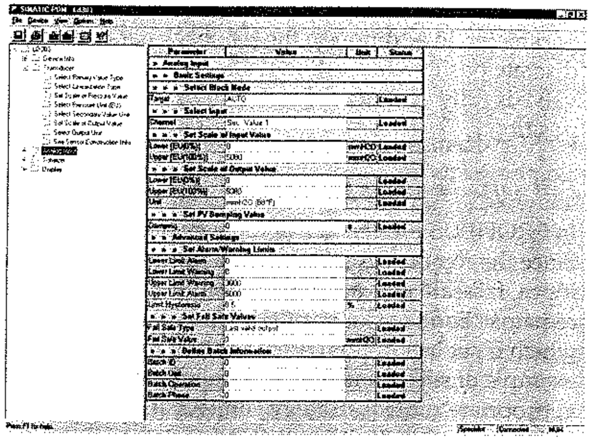
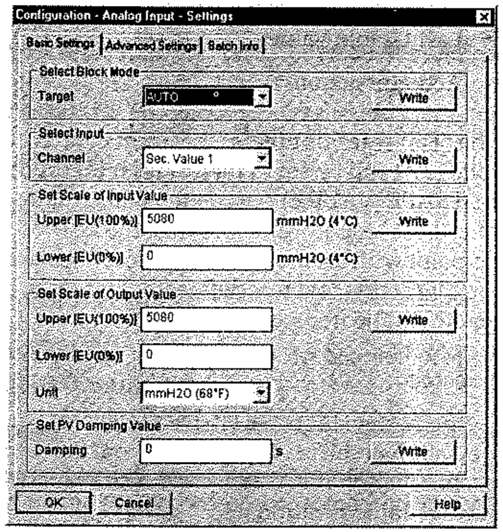


Figure 4-18. AI block overview.



Select Block Mode
Target: AUTO

Select Input
Channel: Sec. Value 1

Set Scale of Input Value
Upper [EU(100%)]: 5080 mmH2O (4°C)
Lower [EU(0%)]: 0 mmH2O (4°C)

Set Scale of Output Value
Upper [EU(100%)]: 5080
Lower [EU(0%)]: 0
Unit: mmH2O (68°F)

Set PV Damping Value
Damping: 0 s

Figure 4-19. AI block configuration.

where the readback values comes from. Rather than setting output action in IO_OPTS, it is set from the INCREASE_CLOSE parameter. In addition to the readback parameter, there is also a discrete position indication (POS_D) parameter. It indicates if the valve is fully opened, fully closed, or in some intermediate position.

Fail-safe

In PROFIBUS PA, the active fail-safe mode is Auto instead of LO. Since the remote cascade setpoint input (RCAS_IN) is the main input, fail-safe gets activated if the remote cascade setpoint input remains "Bad" for longer than fail-safe time-out (FSAFE_TIME), which is set in seconds. A loss of communication would be an example of such an instance. Similarly, a controller with an appropriate control strategy configuration can initiate fail-safe action by setting "Initiate Fail Safe" status on the remote cascade setpoint input. When the fail-safe is activated the fail-safe type (FSAFE_ TYPE) parameter configures three options regarding what should happen:

- Fail-safe
- Last Usable Value
- Actuator action

For the "Fail-safe" option, in the event of activated fail-safe, the setpoint (SP) will be set to the value configured in the fail-safe value (FSAFE_VALUE) parameter, with a status of "Uncertain." For the "Last Usable Value" option, the last valid value becomes the setpoint with a status of "Uncertain." For the "Actuator action" option, the action taken is defined by the ACTUATOR_ACTION parameter in the transducer block.

Diagnostics

The setpoint deviation (SETP_DEVIATION) parameter indicates the deviation of the readback value from the setpoint in percentage terms. In other words, this is the deviation between the desired and actual valve position.

A summary of the device's current hardware and software error status is kept in the check back (CHECK_BACK) parameter. Checks include the following:

- Fail-safe active
- Request for local Operation

- Field device under local control, LOCKED OUT switch is in gear
- Emergency override active
- Actual position feedback different from expected position
- Torque limit in OPEN direction is exceeded
- Torque limit in CLOSE direction is exceeded
- Travel time for actuator has exceeded
- Actuator is moving toward open direction
- Actuator is moving toward close direction
- Static data changed
- Simulation enabled
- Internal control loop disturbed
- Positioner inactive (output status is "Bad")
- Device under self-test
- Total valve travel limit is exceeded
- Additional input is activated

A configuration tool may use the check back mask (CHECK_BACK_MASK) parameter to filter out the checks that are not performed by the device. This parameter is not really useful for humans.

TOT - Totalizer Function Block

The totalizer function block is very much like an input class block because it receives its input directly from a transducer block over a hardware channel that is selected by the channel (CHANNEL) parameter. The transducer block supports alarm detection (figure 4-20).

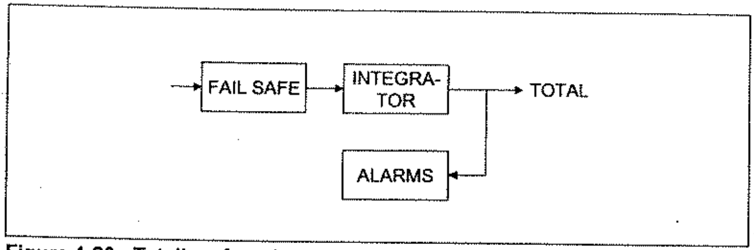


Figure 4-20. Totalizer function block schematic diagram.

The measured rate value received from the transducer block is integrated over time to obtain the total (TOTAL) parameter. For example, if the measured rate value is volumetric flow then the integrated total becomes volume. The behavior of the totalizer block can be configured with the totalization mode (MODE_TOT) parameter. The available options are these:

- Balanced
- Positive only
- Negative only
- Hold

In the "Balanced" mode positive inputs are added, and negative values are subtracted. For the "Positive only" option only positive values are totalized. Similarly, for the "Negative only" option only negative values are totalized. The "Hold" option stops all totalization. When the primary value from the transducer block is "Bad" the fail-safe type (FAIL_TOT) parameter has configured three options as to what should happen:

- Run
- Hold
- Memory

For the "Run" option the totalization continues as if the status would have been "Good." For the "Hold" option the totalization is suspended until the status becomes "Good." For the "Memory" option the totalization continues based on the last usable value. The engineering unit of the integrated value is set in the total unit (UNIT_TOT) parameter. This parameter must be compatible with the transducer block value in the sense that it should be a unit of the same physical quantity as the integrated rate. For example, if the measure value is cm^3/s then the total unit must be a volume unit such as cm^3, dm^3, or m^3 and so on. The accumulated total can be reset to zero or preset using the set total (SET_TOT) parameter. The options are as follows:

- Totalize
- Reset
- Preset

The "Totalize" option is the normal state of operation for the block. The "Reset" option will reset the accumulated total to zero. The "Preset" option will set the total to the value that is configured in the preset value (PRESET_TOT) parameter.

It is a good idea to do totalization in a field device since it will continue even if the communication fails, as long as the device has power.

Control Strategy Configuration

FOUNDATION Fieldbus is also a programming language for building control strategies. An integral part of FOUNDATION Fieldbus is the function block programming language. One of the function block types is the flexible function block. It may be used to incorporate other programming languages, such as any of the IEC 61131-3 languages, into the function block control strategy. The IEC 61131-3 languages includes, for example, ladder diagram, which is popular for logic control strategy configuration.

The FOUNDATION Fieldbus standard defines how function blocks interact with each other. Therefore, blocks can be located in devices from different manufacturers and still propagate parameter values and status seamlessly. As a result, the control strategy can be divided up into smaller pieces, each distributed into a different device handling only part of a loop. Thus, the failure of one device affects only a small portion of the control strategy, leaving a majority of them intact. It is therefore a good idea to distribute controls into the field devices as the fault tolerance increases. However, it is also possible to perform part of the control strategy in traditional centralized controllers if the required function is not available in any of the field devices. If controls are done in centralized processors that handle multiple loops you should use dual redundant processor modules to provide adequate fault tolerance.

As in the past, it is a good idea to be able to isolate the physical device configuration from the control strategy to some degree. However, this is best done in the same, single piece of software because the network, device, and control strategy configuration are intimately related. This will make it easier to validate the devices and control strategy separately. In other words, it is quite beneficial to be able to build the control strategy before you have defined which devices will be used. This gives you greater freedom when you select devices as well as when you choose which device the function blocks will be located in. You can design the control strategy without regard to the type of physical devices used, and you can then assign function blocks to execute in any device that supports the particular function block type. If you also use centralized processors, it is recommended that you also select processors where controls are configured in the FOUNDATION Fieldbus function block programming language. When you do so, the configuration environment remains homogeneous by consistently using a single standard language throughout the entire system.

Configuration software is used to build the control strategy. It also checks the consistency of the configuration before downloading to the devices that execute the blocks. The configuration can be downloaded for an entire fieldbus network in one go, or if a new device has been added to the network, just the configuration for that one device can be downloaded, without affecting the other devices on the bus or disturbing control. Only the part of the configuration that is relevant to a particular device is downloaded to that device.

FOUNDATION function blocks have many parameters, this is so they are flexible enough to be used in basically any application. In most instances, only a few of the parameters are used. Oftentimes, you only have to configure the most common parameters, such as the familiar mode, setpoint, and tuning in the PID block. For a vast majority of parameters, the default values are used. An important feature of a configuration tool is templates with prebuilt device and strategy configurations. Templates simplify the process of verifying the system and make it easier to prove the correctness of the control strategy.

ANSI/ISA-88.01-1995 Modeling

The ANSI/ISA-88.01-1995 is a standard for batch control, but many of the models and terminology are also useful for continuous regulatory control. Instrumentation that is based on fieldbus technologies can be used in both continuous and batch control. For example, the fieldbus devices may execute the phase logic based on the control recipe in the batch management software. Several configuration tools for fieldbuses make use of the ANSI/ISA-88.01-1995 models. Moreover, the PROFIBUS PA specification has a universal function block parameter for batch identification that is based on the ANSI/ISA-S88.01-1995 procedural control model.

The physical model deals with the way the processing equipment like plant machinery and vessels are hierarchically structured. An area consists of one or more process cells and so on down to the control modules (figure 4-21). The control module is the level that is most relevant to function blocks in fieldbus devices such as transmitters and valves, which make up simple control activities like PID control loops that "connect" to the process. The unit and the equipment modules are the most relevant physical levels for the procedural control in the batch management software.

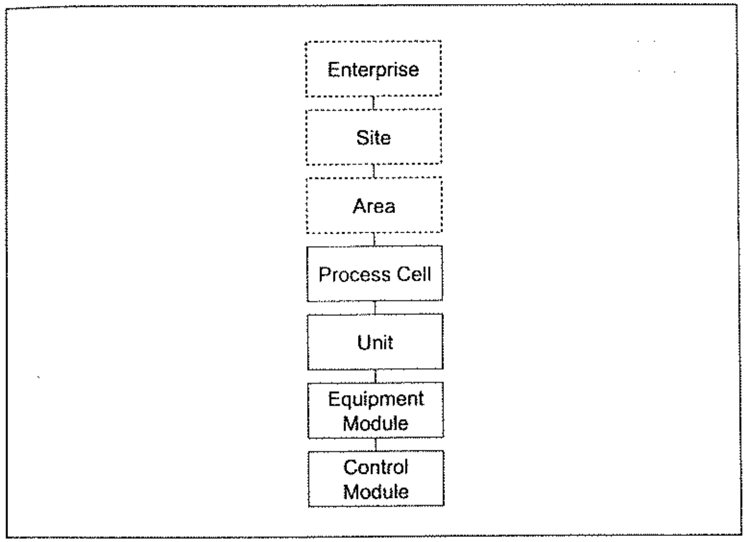
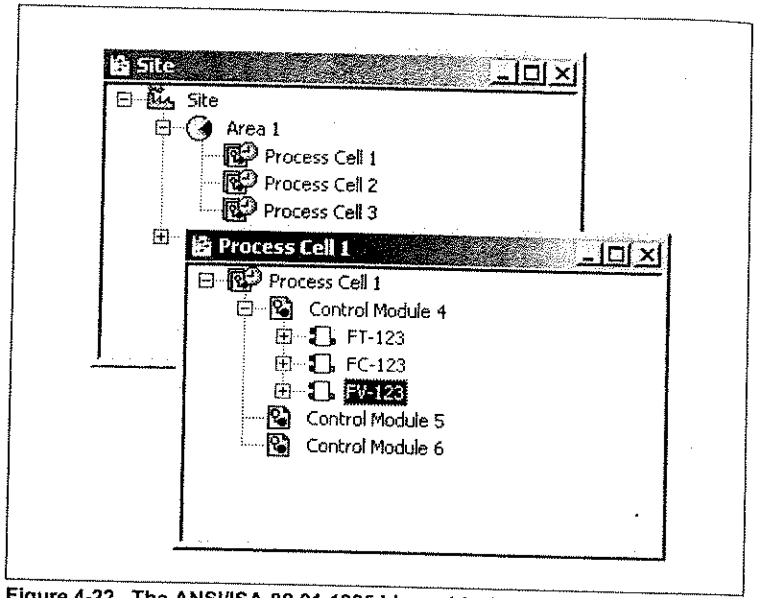


Figure 4-21. The ANSI/ISA-88.01-1995 physical model for processing equipment.

The process cell is different from any other ANSI/ISA-S88.01-1995 abstraction in that it is a clear boundary where there is no or minimal interaction between process cells. A process cell also encapsulates the related control activities. A configuration tool using ANSI/ISA-S88.01-1995 forces plants to use a modular approach to build structured control strategies, which make the system more robust (figure 4-22). Within a process cell there can be any kind of linkage, for example, between control modules. That is, a function block in one control module can link to a function block in another control module as long as both are within the same process cell. Although it would be tidy to have only the instruments on the same vessel on one network, it would be uneconomical to connect just a few devices to a bus. The configuration tool can isolate the control strategy from the fieldbus networking. That is, the control strategy is organized according to the processing equipment it controls, not the way the instruments are wired together. A control module contains the function blocks for the flow, level, temperature and pressure loop, for example, a basic PID, cascade of two PIDs, or a ratio loop plus alarms and totalization associated with the loop.



Site

Site
Area 1
Process Cell 1
Process Cell 2
Process Cell 3

Process Cell 1

Process Cell 1
Control Module 4
FT-123
FC-123
FV-123
Control Module 5
Control Module 6

Figure 4-22. The ANSI/ISA-88.01-1995 hierarchical structure is used to organize the control strategy.

Simplify the process of navigating the control strategy by naming the process cells and control modules according to the processing equipment (machines and vessels) they control. It may be a good idea to use the strategy (STRATEGY) parameter to store a number that uniquely identifies the control module in which the block is used and to use the alert key (ALERT_KEY) parameter to store the process cell number. This makes it easier for external software to identify where individual blocks belong and where alerts come from.

FOUNDATION Fieldbus Programming Language

The function block programming language is an integral part of FOUNDATION Fieldbus. It is ideal for building strategies for regulatory control and monitoring. However, several blocks are available for discrete logic functions. By selecting, linking, and parameterizing these blocks you can make control strategies. The transducer, resource, and function blocks work the same way in H1 and HSE devices.

Terminology and Basics

Several types of standard function blocks created by the Fieldbus Foundation are available to perform the various functions required in a control system, both analog and discrete. By using the right blocks, users can build basically any control schemes. Manufacturers also create their own function blocks to provide special functions. The control strategy configuration is rather detached from the device since the function blocks can be executed in any one of several devices, although eventually every block has to be assigned to some device. The control loop spans many devices and perhaps more than one network. The function blocks have many characteristics in common with the resource block and transducer blocks used for individual device configuration.

Function Blocks

Each type of function block has a different internal algorithm and several parameters that perform different types of functions. Function blocks perform the basic monitoring and control functionality independently of I/O hardware. For example, the analog input (AI) block provides the basic functionality needed for measurement: simulation, inferential range, transfer function, damping, and alarm. The standard AI block in a pressure transmitter is the same as that of a temperature transmitter. The standard PID block is the same whether it is in a transmitter, positioner, or centralized controller, independent of manufacturer. There are four subclasses of function blocks with different characteristics (figure 4-23):

- Input Class
- Control Class
- Calculate Class
- Output Class

The input class blocks connect to sensor hardware via an input transducer block over a hardware channel. Control class blocks perform closed-loop control and have back-calculation functionality to provide bumpless mode transfers and reset windup protection, among other features. The output class blocks connect to actuator hardware via output transducer blocks over a hardware channel and support the back-calculation mechanism. Calculate class blocks perform auxiliary functions required for control or monitoring, but they do not support the back-calculation mechanism.

A large number of standard function blocks have already been defined by the Fieldbus Foundation. In addition to these, manufacturers define specific blocks that may be extensions of standard

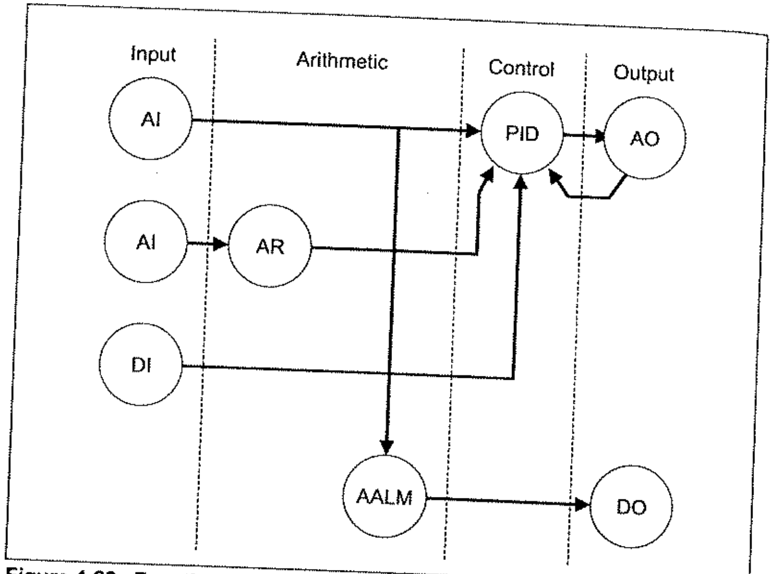


Figure 4-23. Function block classes.

blocks or completely unique. Standard and manufacturer-specific blocks can be linked together because the input and output parameters always follow the specification, that is, they are interoperable. A host tool is able to configure manufacturer-specific blocks because their parameters are defined in the device support files.

The type of blocks available in a device and the number that can be used simultaneously differ from one manufacturer and device type to another. Some devices may have a fixed set of blocks, for example, one each of three to four types. Advanced devices have dynamically instantiable blocks. This means that they have a large library of block types, and users can select as many blocks of the same or different type as the memory can hold. The configuration tools that support the dynamic instantiation of function blocks ensure that only supported blocks are allocated to a device and that memory and other device resources are sufficient. For example, a device can have several PID blocks. The dynamic instantiation capability makes it easier to use the device to replace a failed device, since there is a greater chance that the new device can be instantiated with the blocks used in the previous device. Other advantages of dynamic instantiation include greater flexibility for building control strategies and applications of the device.

It may be a good idea to use the standard FOUNDATION function blocks in the control strategy insofar as possible since this will make it easier to replace the device in the future without having to modify the control strategy. The function block can simply be reassigned, usually as a simple drag-and-drop operation, to the new device since most devices do support the standard blocks. This also achieves interchangeability. Devices that have dynamically instantiable function blocks usually support both the standard blocks and the enhanced versions, for example, the standard PID block plus an advanced PID block with additional functions. Configuration tools generally have a library of preconfigured control strategy control modules that use the standard blocks. This allows the templates to be assigned to a wide range of devices.

Function Block Parameters

The resource block, transducer blocks, and function blocks all have parameters contained in them to configure and operate the block as well as to do diagnostics. Function blocks also have input parameters that are processed by the block algorithm, producing output parameters. In all, there are three types of parameters in a function block:

- Contained parameters
- Input parameters
- Output parameters

For example, in a PID block the process variable goes to one of its inputs, the manipulated variable is one of the outputs, and the tuning parameters are some of the contained parameters. Parameters can be set by the user or by the block itself. Input parameters generally get their value from an output that is linked or set by the user, but the block itself cannot set its input values.

Function Block Links

Function blocks are linked to each other from output parameters to input parameters. The link carries both the value and the status of the parameter. An output parameter can link to any number of inputs. Links between function blocks in different devices are communicated over the network. Links within the device do not have to be communicated on the bus. They are therefore immediate and do not take up network bandwidth (figure 4-24). As a result, it is desirable to reduce the number of communications between devices. This can be done by allocating the function blocks to the devices in such a way that as many of the links as possible are kept internally, thereby improving loop response time. The resource block and transducer blocks are not part of the control strategy, and all their parameters are contained and cannot be linked.

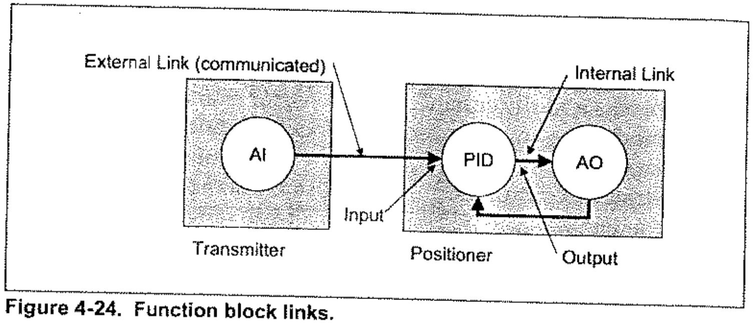


Figure 4-24. Function block links.

An input parameter may also be linked to another input parameter but only within the same device. It is customary to link inputs instead of linking the same output from one device to two or more inputs in another device; this reduces the number of external links. The configuration tool generally checks the control strategy for unnecessary external links and converts them to internal links. All output parameters with external links are "published" on the network, meaning that the output is available to any input that wants to use it. Input with an external link "subscribes" to the respective output. If communication fails to update an input parameter this is also reflected in the status, which allows the block to take action and bring it to the attention of the operator. Discrete outputs can only be linked to discrete inputs. Likewise, analog outputs only link to analog inputs. The user may write input parameters that are not linked, but input parameters that are linked may not be written. There are three types of links:

- Noncascade (forward)
- Cascade forward
- Cascade backward

In fieldbus terminology, the block that is the source of a forward link is said to be the "higher" or "upstream" block, and, consequently, the block that is receiving the forward link is said to be the "lower" or "downstream" block. The term *cascade* applies not only to the classical control strategy where the output of a primary "master" PID controller becomes the setpoint of a secondary "slave" PID controller. *Cascade* is here used in a wider sense for any type of function block that receives a setpoint from another function block. For example, the output (OUT) of a PID block is linked to the cascade input (CAS_IN) and becomes the setpoint of an analog output (AO) block and then subsequently as well for the servo that controls the position of the control valve. Though this broader meaning of *cascade* may appear strange at first, it soon makes sense. Associated with the forward cascade link that brings the output from the upstream block to the cascade setpoint of the downstream block, there is a backward feedback link from the lower block back up to the setpoint source. The backward link goes from the back-calculation output (BKCAL_OUT) to the back-calculation input (BKCAL_IN) and is used to provide a number of useful interlocks and bumpless transfer. The forward and backward cascade links are together called a cascade structure (figure 4-25).

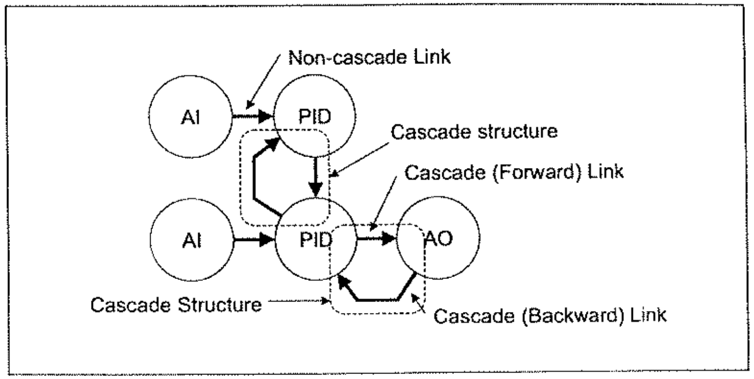


Figure 4-25. Link types and cascade structure.

Example: A basic PID loop is configured using three blocks: analog input (AI), PID control (PID), and analog output (AO). Three function block links are required (figure 4-26).

The first link is from the AI block output (OUT) to the PID block primary input (IN) for the process variable. The second is from the PID block output (OUT) to the AO block cascade setpoint input (CAS_IN). Lastly, there is a link from the AO block back-calculation output (BKCAL_OUT) backward to the PID block back-calculation input (BKCAL_IN). Thus, the cascade structure between PID and AO is the same as between two PIDs (figure 4-27).

Back-calculation outputs, and only back-calculation outputs, should be linked to back-calculation inputs. A back-calculation

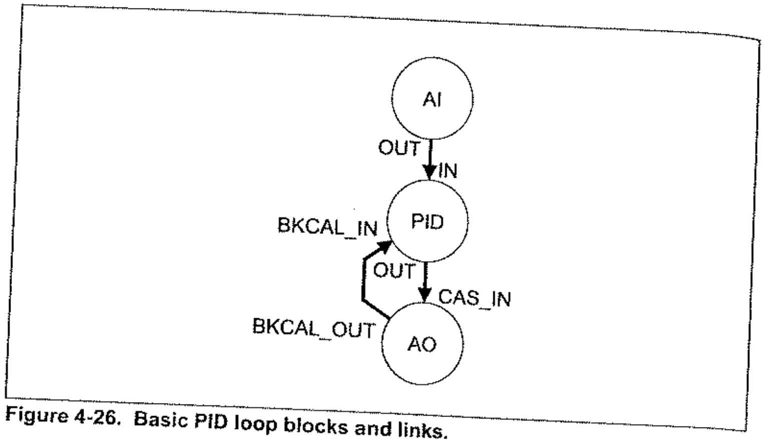


Figure 4-26. Basic PID loop blocks and links.

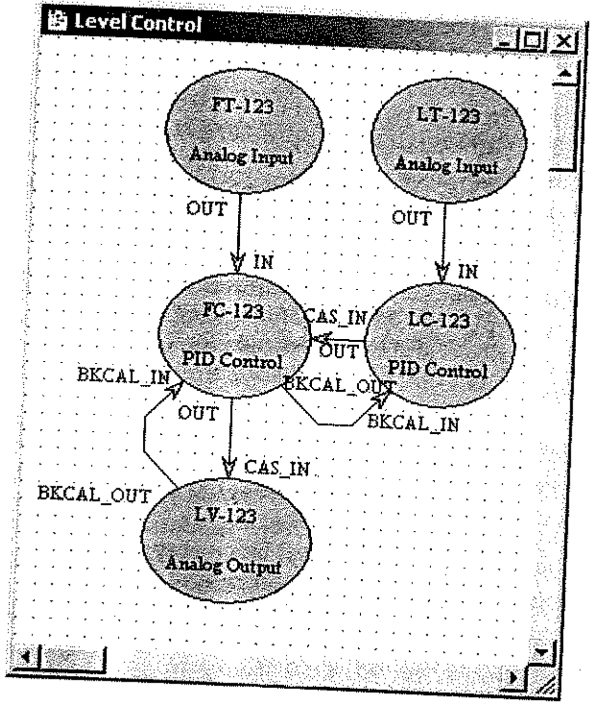


Figure 4-27. A cascade loop has two cascade structures, one between the two PID blocks and the second to the AO.

output should be linked to one, and only one, input. A discrete output should only go to discrete inputs, and an analog output only to analog inputs.

Function Block Interlocks

A linked output parameter value is propagated together with the status to the input parameter of the receiving block to inform whether a value is suitable for control. It may also function as a feedback informing whether the output does not move the final control element, and so on. Status is used for several built-in interlock functions. For example, if a sensor fails, the AI block status will inform the PID block, which will stop controlling. If a control valve is hand-operated, the AO block feedback link status will inform the PID block, which will then initialize its output to prevent reset windup and later provide bumpless return to automatic. It is therefore a good idea to use FOUNDATION Function blocks throughout the control strategy, without other intermediate languages. This will enable you to fully benefit from this built-in functionality rather than having to implement and verify discrete logic.

Function Block Execution

Function blocks receive inputs and execute their algorithm to produce an output that is passed to the next block. There are three ways in which execution of function blocks may be coordinated:

- Scheduled
- Chained
- Manufacturer specific

The function blocks are typically executed according to a schedule prepared by the configuration tool. The schedule indicates at what instant each function block shall be executed and each link communicated. For example, a simple PID loop would start with the execution of the AI block in a transmitter followed by the communication of the external link from the AI block output to the input of the PID block in a valve positioner. The PID block would then be executed, followed by the AO block in the same device. The function blocks are executed over and over again, typically several times per second. Since the blocks are distributed over several devices, parallel loops are executed simultaneously in true multitasking fashion. The period with which the function blocks on a network are executed is called a "macro cycle." The resource block and the transducer blocks are not part of the control strategy, and therefore their execution is not controlled by the schedule. Rather, their execution is unique to the device. For the chained method, the function blocks within a device are executed one after the other based on when the block execution is completed.

Common Characteristics for Function Blocks

Each function block must be given a function block tag (FB_TAG) to identify it. The tag must be unique in the system, and it may be up to thirty-two characters long. For easy cross-referencing, you should use the same tags as are used in the P&I diagrams in the control strategy configuration. It may be a good idea to use physical device tags and function block tags based on ANSI/ISA-5.1-1984 (R1992) - Instrumentation Symbols and Identification. For example, you might use "10-PT-2308" for an analog input block, meaning process cell 10, pressure measurement, loop (control module) 2308. The different function block types have been implemented consistently, and therefore many block types have many parameters and characteristics in common. This is particularly true for function block types within the same class: input, control, calculate, and output.

Alarm and Event Parameters

Alarms and events are collectively called "alerts." Alarms occur and remain active for some time, after which they are clear. Events, on the other hand, only occur. Alarms are reported both when they occur and when they clear. Since events do not remain active they are never reported as cleared. An example of an event is a parameter change. Alarms and events can be configured to be detected and time-stamped in the field devices, provided the host supports the acceptance of the associated alerts reported by the field devices. The parameters used to configure the alarms and check alarm status are the same across the range of blocks. Depending on the block type, the type of standard alarms may vary. The standard blocks may have some of the following alarms:

- Write enable (WRITE*)
- Discrete (DISC*)
- Process variable high-high (HI_HI*)
- Process variable high (HI*)
- Process variable low-low (LO_LO*)
- Process variable low (LO*)
- Deviation high (DV_HI*)

- Deviation low (DV_LO*)
- Block diagnostics (BLOCK_ALM)

Most alarms have three associated parameters:

- Trip limit value (*_LIM)
- Priority (*_PRI)
- Alarm status (*_ALM)

The trip limit parameters have names ending with _LIM. The setting of the trip value is restricted by the applicable scaling parameter, for example, the output scale in case of the AI block and the process variable scale in the case of the PID block. The default values for the alarm limits are negative infinite (-Inf) for low alarms and positive infinite (+Inf) for high alarms, which effectively prevents them from occurring. The alert priority parameters have names ending with _PRI for configuring the alarm priority level. The parameter names ending with _ALM contain the status information for the latest occurrence of the alarm.

Manufacturer-specific blocks have alarms that are implemented in a fashion consistent with these parameters. The alarm for change in "Write enable" status is only found in the resource block. The two parameters have names starting with WRITE. The "Discrete" alarm parameters have names starting with DISC. The "Discrete" alarm triggers when the state equals the alarm setting. For the "Block diagnostics" alarm there is a common status parameter for the latest diagnostic alarm (BLOCK_ALM). The block alarm is used by asset management software to track the health of devices.

The parameters for the deviation between process variable and setpoint have names beginning with DV_HI for process variables that exceed the setpoint and DV_LO for when the process variable is below the setpoint, regardless of control action. The parameters for the four process variable alarms have names starting with HI_HI for the process variable that exceeds the critically high limit, HI for exceeding the advisory high limit, LO_LO when the process variable is below the critically low limit, and LO for advisory low limit. The parameters are summarized in table 4-1.

The "Write enable" alarm is triggered when the write-protection status is changed and therefore does not need any adjustable trip limit. It makes sense to enable this alarm for devices that are write protected. The "Discrete" alarm trips when the associated value is the same as the set limit. The "Process variable" and deviation

Table 4-1. Standard alarm parameter names.

Description	Trip Limit	Priority	Status
Write enable		WRITE_PRI	WRITE_ALM
Discrete	DISC/LIM	DISC_PRI	DISC_ALM
Process variable high-high	HI_HI_LIM	HI_HI_PRI	HI_HI_ALM
Process variable high	HI_LIM	HI_PRI	HI_ALM
Process variable low-low	LO_LIM	LO_PRI	LO_ALM
Process variable low	LO_LO_LIM	LO_LO_PRI	LO_LO_ALM
Deviation high	DV_HI_LIM	DV_HI_PRI	DV_HI_ALM
Deviation low	DV_LO_LIM	DV_LO_PRI	DV_LO_ALM
Block diagnostics			BLOCK_ALM

alarm limit values are configured in the same unit as the process variable. The block alarm is triggered whenever the device diagnostics detect a fault in the block. You should configure the priority for each alarm to let the host arrange the alarm reporting based on priority. This will make the job easier for the operator. It is a good idea to use these same priority levels throughout the system for consistency and to eliminate mistakes, that is, use them also for alarms that are detected in a shared central controller or host. The alert priority shall be configured as follows:

0	Not detected
1	Detected but not reported to host
2	Requires no operator attention
3-7	Advisory alarms
8-15	Critical alarms

The read-only alarm status parameter includes elements for the following information about the latest occurring or clearing of the alarm:

- If the alarm has been acknowledged
- If the alarm is still active and if it has been reported to the host
- Date and time stamp of occurrence/clear
- Type of alarm (only used for block diagnostic alarm)
- Actual value that tripped or cleared the alarm

To enable the time stamping of alerts at the source, that is, in the field device, each device has a built-in clock that is regularly synchronized by a time master, which is usually in the host.

The process variable and deviation alarms have a common hysteresis parameter (ALARM_HYS). It is expressed as a percentage of the range set in the process variable scale (PV_SCALE) of the block but in the output scale for the AI block. The hysteresis makes sure that the alarm does not chatter (rapidly activate and deactivate) when the value is close to the alarm trip limit. It does this by making the deactivation value different from the activation value by the amount specified (figure 4-28). A high alarm is activated when the associated value is higher than the limit and deactivated when the value is lower than the limit, minus the alarm hysteresis. Likewise, a low alarm is activated when the associated value is lower than the limit and deactivated when the value is higher than the limit plus the alarm hysteresis. When the value is within the band set by the hysteresis, the alarm will remain in the "last state" being active or inactive. The hysteresis can be set within the range of 0 to 50%, with 0.5% being the default value.

For example, if the process variable scale is set from -50 mmH2O to +350 mmH2O, 0.5% hysteresis would correspond to 2 mmH2O. If the high alarm is set to trip at 300 mmH2O it will clear at 298.

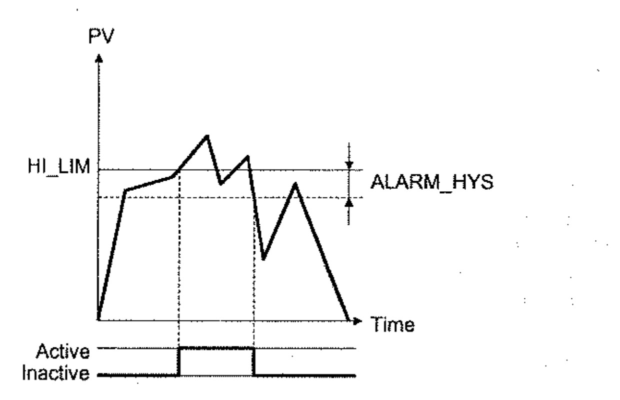


Figure 4-28. Alarm hysteresis on high process variable limit.

By default, the operator may manually acknowledge an active alarm to indicate to alarm management that it has been noticed. Use the auto-acknowledge option (ACK_OPTION) parameter to configure individual alarms in the block to be acknowledged automatically. The operator acknowledgement status does not affect the operation of the block but is indicated in the process variable status. Setting the mode out of service acknowledges all alarms. The acknowledgement of the following alarms can be configured to be automatic:

1. Write enable/Discrete
2. Process variable high-high
3. Process variable high
4. Process variable low-low
5. Process variable low
6. Deviation high
7. Deviation low
8. Block diagnostics

A summary of the current status in the block of all alarms is kept in the alarm summary (ALARM_SUM) parameter. A host may be configured to use it to give the operator an overview of the alarms in the block. Four elements of the parameter track the following information for each alarm:

- Active
- Unacknowledged
- Unreported
- Disabled

The "Active" element indicates which alarm conditions are currently true. The "Unacknowledged" element indicates which alarms have not been acknowledged by the user, that is, perhaps have not yet been noticed by the operator. The "Unreported" element indicates which alarm trip or clear has not yet been successfully transmitted to the host. At the "Disabled" element the unwanted alarms can be disabled. This is the only configurable element in the alarm summary parameter.

The event update (UPDATE_EVT) parameter contains information about the latest write to a static parameter: if it has been reported, if it is acknowledged, which parameter was changed, and the time stamp. Since it is an event it does not remain active and is never

cleared. This parameter can only be read. The alert associated with this parameter has a fixed priority of 2, which is usually configured to not notify the operator. The event update parameter has elements that are similar to the alarm status (_ALM) parameters only that instead of type of alarm and value, it reports the index of the parameter that was changed as well as the static revision (ST_REV), that is, the total number of changes made to the block.

The block alarm status (BLOCK_ALM) parameter is associated with diagnostic errors. The associated faults are those listed for the universal block error (BLOCK_ERR) parameter, which summarizes the diagnostic faults that are present. The information in the elements of the block alarm status parameter is the same as the information of other alarm state (_ALM) parameters. The subcode element informs users what fault the diagnostics detected. The alert associated with this parameter has a fixed priority of 2, which is usually configured to not notify the operator.

Access Permission

The access permission (GRANT_DENY) parameter is available in most function blocks, but it does not affect the block operation. The use of this parameter depends entirely on the configuration of the host, and in most cases it is not used at all. Generally, the controls are supervised by operators from the host workstations who handle such operations as setpoint, output and mode changes, tuning and alarms, and the like. In some applications, particularly in batch control, an automated batch program or other application may handle these functions instead. Likewise, the process may be operated from a local panel. If they wish, engineers can also implement applications such as an automated batch program to utilize the access permission parameter so as to restrict or monitor manual operator intervention to operation, tuning, and alarm parameters in the block. The access permission parameter has two elements, "Grant" and "Deny." The block itself does not use the access permission parameter. All the block does is set the corresponding "Denied" item in case the user removes the "Grant" for one of the parameter sets. In other words, the block "tattles" on the user and exposes any user intervention, by telling the batch program that the user has intervened. There are four sets of parameters associated with the access permission. The number of parameters in each set depends on the parameters available in the block:

- Program (operation parameter access by other applications)
- Tune (tuning parameter access by other applications)

- Alarm (alarm parameter access by other applications)
- Local (operation parameter access by local operator interface)

The operator may grant control of certain sets of block parameters to other applications and local operator interfaces by enabling access in the "Grant" element. If the batch program is implemented as intended it would, as an example, only operate, tune, or manage alarms in the block if the operator has granted access. In that case, the operator should not be able to access the same set of parameters. The operator should also be prevented from accessing the "Denied" element, for example,. by not displaying it on any screens in the process visualization application. By checking the "Denied" element, the batch program, in this example, can track whether or not the user has reclaimed operation (by removing the grant) and intervened during the batch run.

Scaling

In most blocks, the analog parameters are associated with scaling parameters that have names ending with _SCALE. Control, calculate, and output class blocks generally have a process variable scale (PV_SCALE) parameter for the input. Input, control, and calculate class blocks typically have an output scale (OUT_SCALE) parameter for the output. Input and output class blocks have a transducer scale (XD_SCALE) parameter for the value on the I/O hardware channel. Because the blocks operate with values in engineering units rather than scaled in percentages, in many blocks the scaling parameters have no other function than to select the unit. In other blocks, only the range values are used, and the unit has no function. The scaling parameters have four elements:

- Upper Range Value (EU_100%)
- Lower Range Value (EU_0%)
- Unit
- Number of decimal places to be displayed

The host may also be configured to use the scaling information to display bargraphs and animate filling tanks, among other activities. In some blocks, the operational range has to be set using the scaling parameter. The most common case is to set the control range for the PID block. Another case in which range has to be set is in the AI block for inferred measurement applications, such as when using a differential pressure transmitter for flow measurement.

Scaling often imposes a restriction on the values they are associated with. For example, the PID alarm and setpoint limits (*_LIM) cannot be set more than 10 percent outside the range imposed by the process variable scale (PV_SCALE) parameter. In the AI block, the same applies but the range is based on the output scale (OUT_SCALE) parameter. The block diagnostics check for configuration errors and report a block alarm for any inconsistencies such as limits that are outside the scale.

Scaling is not passed between function blocks with the link. Thus, there is no requirement for the units or scale in the source and receiving block to match as far as block operation is concerned. However, input and output class function blocks usually must have transducer scale units that match their respective transducer blocks. Therefore, the device typically ensures that the unit set in the transducer scale is passed to the transducer block to ensure configuration consistency and avoid a block alarm.

State Templates

Discrete parameters are associated with state parameters that have names ending with _STATE. The function block does not use the state parameter. A discrete parameter may have 256 valid states. A host may be configured to use the state parameter as a pointer to, for example, a predefined table of messages and colors for each state of the associated parameter.

Parameter Status

Input and output parameters always have an associated status. Contained parameters, for example, process variable (PV) and setpoint (SP), also have status, although there are only a few that do. It should be noted that in most blocks the process variable (PV) parameter and setpoint (SP) parameter may be following the primary input (IN) and cascade input (CAS_IN) parameters, respectively. Whether parameters are actually following these inputs depends on the configuration and mode of the block.

In the section of this chapter on device configuration we discussed the "Bad" and "Uncertain" conditions for the quality portion of the status element. In addition to that portion, the limit part of the status element indicates whether the associated value is limited or constant, or has no constraint. The following limit conditions may be seen:

- None
- Limited High

- Limited Low
- Constant

Generally, the limited condition will be "None," meaning the value is not being limited. "Limited High" means the value has reached its upper limit and cannot go any higher. Consequently, "Limited Low" means the value has reached its lower limit and cannot go any lower. "Constant" means the value cannot go higher or lower. The limit condition is primarily seen in the status of the backward path of a cascade structure for a lower block. There, it tells the upstream block that its setpoint has reached its limit and that the upstream block should not move its output any further in that direction.

There are two forms of the "Good" quality part of the status element. Input and calculate class block outputs use "Good (Noncascade)," whereas control and output class blocks use "Good (cascade)." By looking at the type of "Good" at the input status, a block can determine whether the upstream block is a control block or not. For example, this would allow the PID block to tell if the cascade setpoint input comes from a cascaded control block in which bumpless transfer is possible or from a calculate block where it is not (figure 4-29).

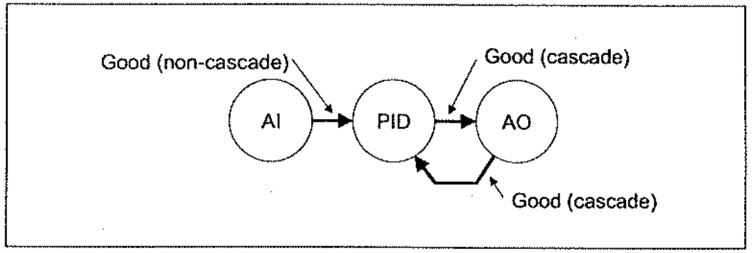


Figure 4-29. Different "Good" identifies an output as capable of cascade initialization or not.

The quality "Good (Noncascade)" means that the value may be used for control. This status is only associated with variables that are not in a cascade structure. Generally, this status comes from input or calculate class function blocks. For the process variable parameter, the substatus will contain information about alarm conditions. A host may be configured to annunciate the alarm in a regular graphic screen by using different colors for the value, based on the status. The subquality for "Good (Noncascade)" can be associated with the following reasons:

- Nonspecific
- Active Block Alarm
- Active Advisory Alarm
- Active Critical Alarm
- Unacknowledged Block Alarm
- Unacknowledged Advisory Alarm
- Unacknowledged Critical Alarm

For "Good (Noncascade)," the subquality "Nonspecific" means there is no alarm. The quality "Good (cascade)" means that the value may be used for control. This status is only associated with variables that can participate in a cascade structure. It generally comes from a control class function block or an output block. The substatus contains information for establishing and initializing cascade structures (figure 4-30). The subquality for "Good (Cascade)" can be associated with the following reasons:

- Nonspecific (forward and backward paths)
- Initialization acknowledgement (forward)
- Initialization request (backward)
- Not invited (backward)
- Not selected (backward)
- Local override mode (backward)
- Fault-state active (backward)
- Initiate fault-state (forward)

Many of the subqualities for "Good (Cascade)" do not have very much meaning to the user. Their main function is to serve as part of the automatic interlocks between the blocks. However, the subquality can be helpful for troubleshooting and untangling some interlocked situations. The function of each subquality for "Good (Cascade)" is intimately linked to the block mode (MODE_BLK) parameter and is therefore explained as part of each mode.

"Bad" status has higher priority than "Uncertain." Therefore, if both bad and uncertain conditions exist the status will be "Bad."

When a linked input is not updated within a certain time because of communication failure its quality will be "Bad" or "Uncertain," depending on the block.

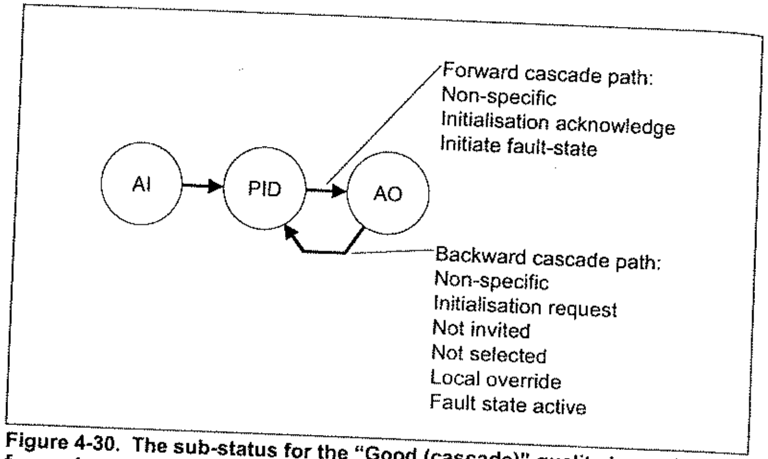


Figure 4-30. The sub-status for the "Good (cascade)" quality is used in the forward and backward cascade.

Along with the value from output parameter, the status is carried to the input parameter of the next block. The backward path limit and constant information is used, for example, to prevent windup in the PID integrator. Status is also used to provide bumpless transfer and initialization between blocks. If the block has failed, the outputs are "Bad" and are not used by other blocks. Limit and initialization information is carried from lower to higher blocks through the backward path. The status of the input may change the mode of the block. For example, in a control class block a "Bad" input may result in manual mode. Calculate class blocks normally propagate limit and quality information from their inputs to their output, that is, from higher to lower blocks.

When the output of an upper block reaches the limit for the setpoint in a lower block, the lower block will set limited status in the back-calculation output to the upper block. This will prevent the upper block from moving its output further in that direction, but the upper block will not change mode.

Fault state

Fail-safe action is implemented in the output class function blocks to shut down the basic controls when a failure is detected somewhere in the loop. It is called fault state rather than fail-safe in the FOUNDATION Fieldbus programming language because it is a state requested by the strategy, but ultimately it is the hardware that has to ensure it happens. The control strategy makes a best effort, but the function and success depend on many factors. In the presence of certain faults such as power loss (electrical, pneumatic, hydraulic, etc.) the fault state cannot take effect, and ultimately the hardware must also be implemented to fail safely in its deenergized state using traditional methods such as spring return. In this book, the terms *fail-safe* and *fault-state* are used interchangeably, although the subtle difference in meaning should be kept in mind.

In case the input is "Bad" a control class block will optionally set the status option parameter and pass a status to the output class block. This will force the fail-safe state to bring the loop to a graceful shutdown because control cannot continue without input. Similarly, the fault state is activated if the remote setpoint input for the output class blocks are not being updated due to a loss of communication. The output blocks may be configured to freeze in the last position or go to a predefined safe value. The fault state may also be forced from the resource block. It is a good idea to make use of this feature of the FOUNDATION Fieldbus programming language because it eliminates the need to configure and verify shutdown interlocks for the basic controls using discrete logic.

Whether the shutdown functionality provided by the basic controls is sufficient or additional SIL-rated equipment is required must be decided by doing a safety assessment of each loop.

Status Options

Most function blocks have a status options (STATUS_OPTS) parameter by which the block behavior can be configured in response to different status conditions. This parameter can only be set when the block is in OOS mode during online configuration, but there is no such restriction when configuring off line. The options that can be set are as follows:

- Initiate fault state if input (IN) is "Bad"
- Initiate fault state if cascade input (CAS_IN) is "Bad"
- Use "Uncertain" as "Good"
- Propagate fault forward
- Propagate fault backward
- Set Target mode to manual if input (IN) is "Bad"
- Set quality as "Uncertain" if limited
- Set quality as "Bad" if limited
- Set quality as "Uncertain" if block is manual mode

Control class blocks do not propagate status from higher to lower blocks. For example, in the PID block a "Bad" input from AI block or back-calculation input does not normally become a "Bad" output; it just puts the PID block in manual mode. Control class blocks may also be configured to set a fail-safe status on the output in the event of a "Bad" primary input (IN). This done by enabling the status option "IFS if BAD IN." This will initiate fail-safe action in the output class block, for example, a sensor failure will close the control valve (figure 4-31).

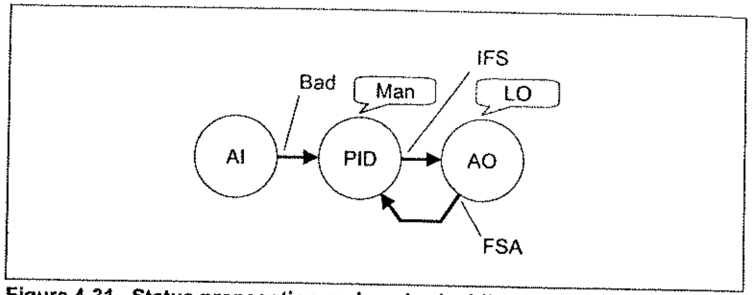


Figure 4-31. Status propagation and mode shedding in response to a sensor failure.

It is a good idea to use this feature if the AI block and PID block are located in different devices, which is most often the case. This will ensure that in the event of a loss of communication the loop can be brought to a safe state.

Similarly, you may configure control class blocks to set a fail-safe status on the output in the event of a "Bad" cascade setpoint input (CAS_IN) by enabling the status option "IFS if BAD CAS_IN." It will in turn initiate fail-safe action in the output class block. For example, normally a secondary PID block would shed from Cas to Auto mode if there is no communication on the cascade setpoint input. If this option is enabled, IFS will be passed to the AO block if the link from the primary PID is lost.

You may configure the status option "Use Uncertain as Good" in control class blocks to treat "Uncertain" as either "Good" or "Bad." By default, "Uncertain" is treated as "Bad." This means that minor problems would be treated just like more serious ones (for example, initiating a shutdown of the loop), which is the safer option. Alternatively, for greater availability you should enable the "Use Uncertain as Good" option so the block continues to operate as if the quality was "Good."

The options "Propagate Fault Forward" and "Propagate Fault Backward" are described in detail in the sections of this chapter on the input and output class blocks, respectively. The overall purpose of these options is to report a fault in the control block they are connected to rather than in the block itself. The fault propagation options are ideal for control loops because they allow a loop to be monitored by looking at a single block rather than at three. This reduces the amount of block information that has to be communicated and consequently improves control performance.

Some control class blocks like the PID shed to an actual manual mode if the primary input (IN) is "Bad." However, the block normally returns to the set target mode, such as automatic or cascade, once the "Bad" input condition has cleared. This allows normal control to resume, thereby maximizing availability. However, for some applications, it may be safer for the loop to remain in manual mode rather than return to the target mode. For this purpose, you can enable the status option "Target to Manual if BAD IN." When you use this option, once the primary input (e.g., from an AI block) goes "Bad," the target mode of the PID will become manual and remain there until it is changed by the operator.

Input and calculate class block outputs are normally "Good" even while the output is limited at one of its extremes. Alternatively, you can configure the block using the status option "Uncertain if Limited" to set the output status as "Uncertain" whenever the output is limited. This would, for example, allow some control blocks to continue normal control (those configured to treat "Uncertain" as "Good") while other control class blocks take fail-safe action. Another option for the input and calculate class blocks is the status option "BAD if Limited." This would cause all control class blocks to initiate fail-safe action. It may be a good idea to use these options for inferred measurements where value conversion or calculation is performed. They will ensure that control is not done based on abnormal values that fall outside the range that has been defined. Similarly, the status option "Uncertain if Man mode" makes the output quality of input and calculate class blocks "Uncertain" if the mode is set to manual. However, this is rarely the case since the blocks are typically left operating in automatic.

Block Mode

The main purpose of the block mode (MODE_BLK) parameter is to determine the source of the block setpoint (SP) and primary output (OUT). These two selections are combined into a single parameter (figure 4-32).

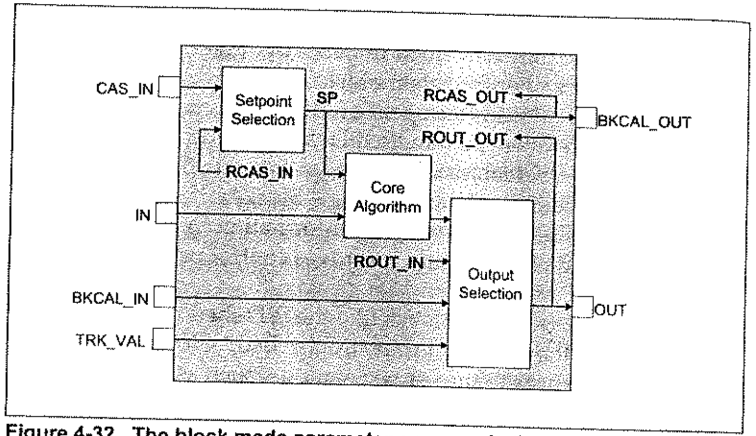


Figure 4-32. The block mode parameter commands the setpoint and output selection and thereby the overall operation of the block.

The source for the setpoint or output may be the operator (through the host), another block, the block itself, or some nonfunction block application software. Depending on the block type, either setpoint or output selection may be unavailable. Essentially, there is no setpoint selection in an input class block and no output selection in an output class block. The complete set of modes and the source for setpoint and output are shown in table 4-2. There are eight modes. For the Actual mode element, only one of the eight prevails at one time:

Table 4-2. Mode setpoint and output source (parameter) summary. In some modes* the setpoint tracks the PV or is maintained in its last selection.

Mode	Setpoint Source	Output Source	Remark
OOS	Operation (SP)	Operator (OUT)	Highest priority
IMan	Operator (SP)*	Lower Block (BKCAL_IN)	
LO	Operator (SP)*	Other Block (TRK_VAL)	Control Class
	Safe Value (FSAFE_VAL)	N.A.	Output Class
Man	Operator (SP)*	Operator (OUT)	
Auto	Operator (SP)	This Block	
Cas	Higher Block (CAS_IN)	This Block	
RCas	Other Application (RCAS_IN)	This Block	
ROut	Operator (SP)*	Other Application (ROUT_IN)	Lowest priority

In addition to the just-mentioned explicit source selections, the setpoint may in certain modes be tracking the process variable (PV) or remain at its last setting. Note that the local override mode behaves differently in an output class block than in a control class block. The main characteristics of the different modes are as follows:

Out of Service (OOS): The block is not being executed. In this mode, the setpoint and output are held at their last value, or, in the case of output class blocks, they may go to the safe state. This is always a Permitted mode that is supported by all blocks. If the resource block mode is OOS, the Actual mode of all the blocks in the device is OOS. That is, the whole device is put in "hold."

Initialization Manual (IMan): An operator cannot select this mode as Target mode. It is activated by a certain status on the back-calculation input (BKCAL_IN) and overrides any mode set by the operator as well as Local Override (LO) mode. The block output (OUT) is initialized to the back-calculation input. This mode is always permitted if it is supported by the block type. The setpoint is determined by the setpoint that is selected or optionally by tracking the process variable.

Local Override (LO) in a control class block (output override): An operator cannot select this mode as Target mode. It is typically activated by the discrete tracking input (TRK_IN_D) in an "output tracking" scheme. This mode overrides any mode set by the operator, except manual. By enabling the track in manual control option the LO mode also overrides Man mode. The block output (OUT) tracks the track value input (TRK_VAL). This mode is always permitted if it is supported by the block type. The setpoint is determined by the setpoint selection or optionally by tracking the process variable.

Local Override (LO) in an output class block (setpoint override): An operator cannot select this mode as Target mode. It is typically activated by a certain status on the cascade setpoint input in a "failsafe" scheme or possibly by a "hand operation" switch on the hardware. This mode overrides any mode set by the operator. The setpoint (SP) is frozen overridden by the fail-safe value (FSAFE_VAL). This mode is always permitted if it is supported by the block type (figure 4-33).

Manual (Man): The operator sets the block output (OUT). The setpoint is determined by the setpoint selection or optionally by tracking the process variable.

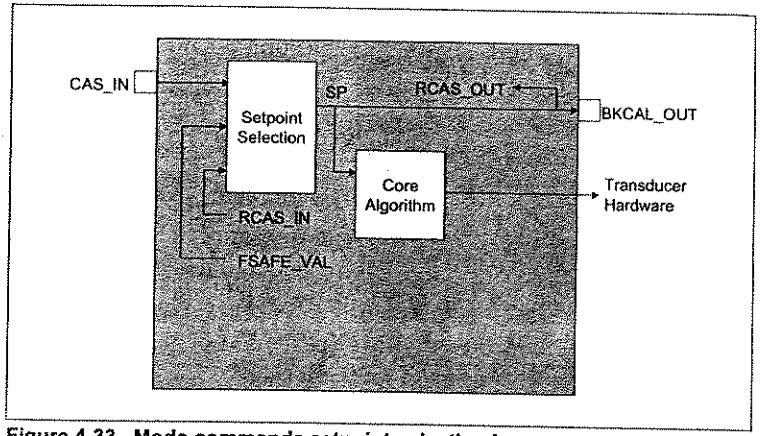


Figure 4-33. Mode commands setpoint selection in an output class block.

Automatic (Auto): The operator sets the setpoint (SP), and the block itself calculates the block output (OUT).

Cascade (Cas): The setpoint is received from another function block through the cascade setpoint input (CAS_IN), and the block itself calculates the block output (OUT).

Remote-Cascade (RCas): The setpoint is received from another application on the remote cascade setpoint input (RCAS_IN). The block itself calculates the block output (OUT).

Remote-Output (ROut): The block output is received from another application on the remote output input (ROUT_IN). The setpoint is determined by the setpoint selection or optionally by tracking the process variable.

The mode parameter is not really used on a day-by-day basis in most blocks. The majority of blocks are set in one operational mode and stay there. The block mode parameter in the PID block is the only one in which the mode is frequently operated as the user sets the loop mode to manual, automatic, or cascade using the mode parameter in this block. Input, calculate, and control class blocks are generally left in automatic mode, and output class blocks are typically left in cascade mode. The remote modes—remote cascade and remote output—are rarely used. However, they are available to enable applications that do not use the FOUNDATION Fieldbus programming language to "link" to the function block that is passing remote setpoints or outputs.

Traditionally, there have been two ways of setting the loop mode from a controller faceplate. One scheme involves three individual buttons for Manual, Automatic, and Cascade mode. The other scheme uses two toggle switches: one that selects the setpoint source as either Local or Remote and one that sets the output source as Manual or Automatic. Although the block mode parameter is geared toward the Manual-Automatic-Cascade scheme a host can be configured to let the user control the mode by using the Local-Remote + Automatic-Manual scheme.

The operator sets one desired operating mode for the block at the Target element of the block mode parameter. The set Target mode may not be achievable during certain error conditions, for example, a process variable sensor failure or when the manipulated variable does not reach the valve. The present mode is seen from the Actual element of the block mode parameter, which can only be read. During the error conditions the Actual mode may change without operator intervention. This is called "mode shedding."

The process engineer may configure the Permitted mode element during initial configuration to enable or disable modes from being selected as Target by the operator during normal operation. The Permitted modes are chosen from the possible modes of the particular block type. Thus, for the Permitted mode element several modes can be chosen. Only Permitted modes may be selected as Target mode during operation. The Normal mode element has no effect on the operation of the block. It may be set by the process engineer and used in a host to remind the operator which mode the loop should returned to for normal operation in the case he or she forgets. Typically, it is the process engineer who selects one mode as normal. Only the Actual and in some cases the Target mode elements are changed by the function block itself, never the Permitted or Normal modes. The block only changes the Target mode if the status option parameter "Target to Manual if BAD IN" has been enabled or the shed option (SHED_OPT) parameter has been configured to allow it to change in Target mode. This feature is used for failure mode shedding when the block must remain in the new mode even after the fault that caused the shedding has cleared.

Typically, the operator will only set Man, Auto, and Cas modes during normal operation. Generally, the operator supervises the control from the process visualization software in the host and sets the mode, setpoint, output, and so on. How the operators interact with the function blocks during normal operation is discussed in detail in chapter 7, "Operation."

There are function block parameters that affect the block output. Therefore, during online configuration these parameters can only be written when the block is in a mode wherein the block itself is not updating the output. This ensures that the process control is not upset by a bump in process variable or controller output. For example, to configure any of the option parameters (*_OPTS) on line you must set the block in OOS mode, and you can only set the scaling parameters (*_SCALE) on line if the block is in manual mode. Of course, the operator cannot set the setpoint if the block is in cascade mode. Similarly, the operator can only set the output if the block is in manual mode. When configuration is performed off line this mode restriction does not apply since the changes are being made to a file, not the operating device.

Cascade Initialization Mechanism

Cascade is the normal operating mode for the secondary PID block in a cascade control loop. Therefore, the setpoint (SP) of the secondary PID is received on the cascade setpoint input (CAS_IN) from the output (OUT) of the primary PID (figure 4-34). However, the secondary PID may be in automatic mode—for example, during startup or other special condition—in which case, the operator instead sets the setpoint directly to the setpoint (SP) parameter. In other words, the output of the primary PID is not used as a setpoint. When the secondary is in Auto the cascade is said to be broken, as there is no path to the final control element.

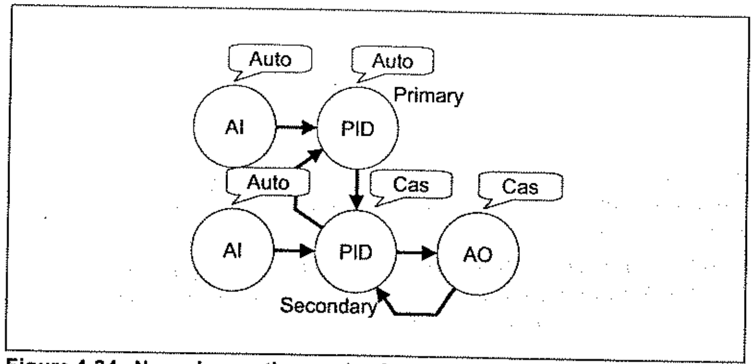


Figure 4-34. Normal operating modes for a cascade loop.

When the mode of the secondary PID is switched from automatic to cascade it is important that the output (OUT) of the primary PID be identical to the setpoint (SP) of the secondary PID. This ensures that the setpoint value does not experience a bump as the mode is changed. For example, if the operator has set a secondary setpoint to 40 percent, the output of the primary PID must also be 40 percent to provide a smooth mode transfer. If the primary PID output had been at some other value, for example, 30 percent, there would be a bump of 10 percent, which would upset the control of the process. The function blocks have a built-in cascade initialization feature to prevent such a bump, that is, to provide bumpless transfer.

The setpoint value of the secondary PID is fed back to the primary PID over the backward path in the cascade structure, that is, the link from the primary PID back-calculation output (BKCAL_OUT) to the back-calculation input (BKCAL_IN) of the primary PID. When the secondary PID is not in cascade mode (for example, in Auto), the substatus of the back-calculation output will be "Not Invited." When the "Not Invited" substatus is received on the back-calculation input of the primary PID block the Actual mode will go into IMan. It will also initialize its output (OUT) to the value on the back-calculation input, that is, the same value as in the secondary setpoint. Consequently, the cascade setpoint input (CAS_IN) on the secondary PID will be identical to its own setpoint (SP) value. As a result, any time the mode in the secondary PID is switched back to cascade there will be no bump in the setpoint value and therefore no bump on the output (figure 4-35). In other words, the cascade structure thus ensures that the process is not perturbed during mode switches.

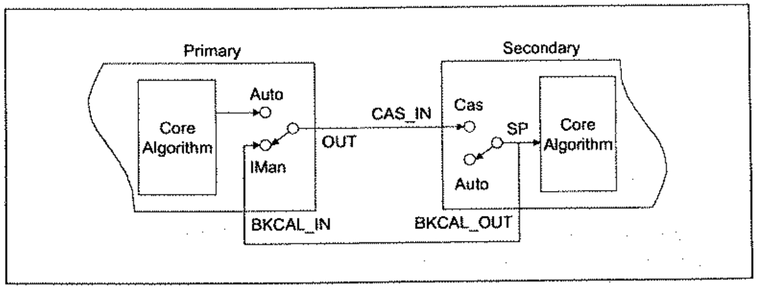


Figure 4-35. To ensure bumpless transfer the secondary setpoint value travels backward to the primary.

Once the Target mode of the secondary PID has been returned to Cas it will pass a "Initialization Request" substatus to the primary

PID block. The substatus of the output (OUT) of the primary PID will be "Initialization Acknowledge." This lets the secondary PID know that it may be switched back to Cas mode without causing a bump in its setpoint because it is equal to the back-calculation input. Mode cannot be switched to Cas unless the initialization in the primary PID block has been acknowledged. When the primary PID block leaves initialization manual the Actual mode will return to the set Target mode. It is sometimes possible to briefly see the "Initialization Acknowledge" on the secondary PID cascade input (CAS_IN) status and the "Initialization Request" briefly on the primary PID back-calculation input (BKCAL_IN) status. However, they are transitional stages that pass very quickly.

In this example, the operator changes the secondary PID block mode by setting the Target to Auto. However, the sequence of events for initialization would be the same in a scenario in which the Target is set at Cas but the Actual mode sheds to some other mode that is not using the cascade setpoint input. Once the condition that caused the shed has disappeared, the block will return to the set Target mode as soon as the initialization of the primary PID has been acknowledged. It is not necessary to remember this sequence since it is completely automatic. However, remember that if the block has back-calculation input, it must be connected to the lower block's back-calculation output, otherwise the cascade will not work. If for any reason (for example, because there is no communication) the back-calculation input (BKCAL_IN) is "Bad" then the mode becomes IMan.

In discrete blocks the corresponding parameters have names ending with "_D". All control class and output class blocks support the same mechanism. in other words, if the mode in the AO block were changed from Cas to Auto the same handshake would take place between the PID and the AO block to ensure a bumpless return to Cas mode.

For the cascade initialization mechanism to work all the blocks from the first PID to the AO, such as selectors and limiters, must support the forward and backward path that makes up the cascade structure. Therefore, only control class and output class blocks can be used in a cascade. For example, the arithmetic (AR) does not support cascade and therefore cannot be connected to the output of a PID. However, for the purpose of adding a feedforward signal to the PID output the PID already has a built-in feedforward function. This, of course, also means that proprietary control strategy programming languages should not be mixed into the same loop since these do not support the safety and bumpless transfer features of the FOUNDATION Fieldbus programming language.

The reset windup protection is propagated backward to the beginning of a cascade. For example, if for some reason the AO block is not in cascade mode it will set "Not Invited" on its back-calculation output. When the secondary PID sees "Not invited" on the back-calculation input it goes into IMan and sets "Not invited" on its back-calculation output, which also puts the primary PID in IMan. The same backward status propagation function also protects the primary PID from reset windup in case the secondary PID reaches a limit. This feature allows control to pick up faster once the secondary PID leaves the limit condition. It is therefore a good idea to use the FOUNDATION blocks at all levels of the cascade to benefit from this protection mechanism.

In the event of an actuator failure the output class block will set a "Bad" status on its back-calculation output. When the control class block receives this status on its back-calculation input, the block will go into IMan mode. This ensures that control using a failed actuator is not attempted and at the same time prevents reset windup (figure 4-36). It is a good idea to use the FOUNDATION blocks throughout the control strategy to benefit from this protection mechanism.

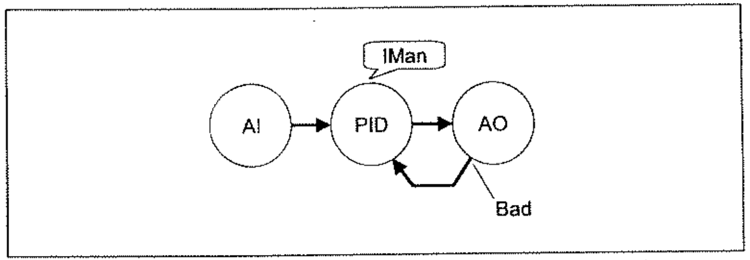


Figure 4-36. Status propagation and mode shedding in response to an output device failure.

Cascade setpoint input need not have an associated back-calculation link if the cascade setpoint comes from an input or calculation class block, that is, if the input has the status "Good (non-cascade)." These blocks have no back-calculation input anyway. For example, the output of an arithmetic (AR) block may be connected to the cascade setpoint input of a PID control block (figure 4-37). Likewise, an AI block can be connected directly to an AO block. The PID block can go into Cas mode and will accept the setpoint directly without any initialization handshake because it can tell from the "Good (non-cascade)" status that initialization is not possible. Of course, this means that switching the PID mode to Cascade will create a bump in the setpoint value.

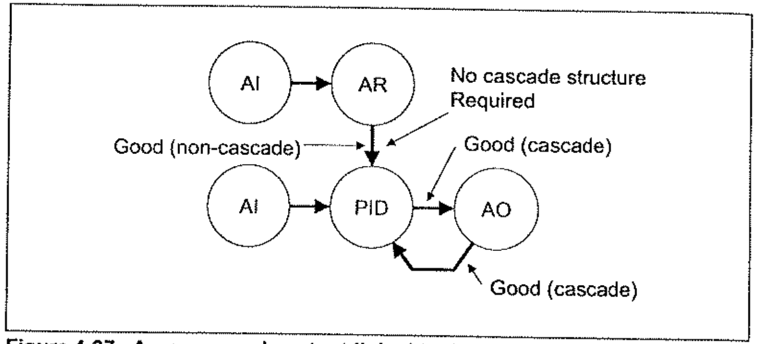


Figure 4-37. A non-cascade output linked to the cascade setpoint input need no associated back calculation link.

Setpoint Limiting

The setpoint (SP), cascade setpoint input (CAS_IN), back-calculation output (BKCAL_OUT), high setpoint absolute limit (SP_HI_LIM), and low absolute setpoint limit (SP_LO_LIM) parameters are set in the same unit as the process variable. In other words, they are set according to the process variable scale (PV_SCALE). In discrete blocks the equivalent parameters have names that end with "_D".

The setpoint by default can be set no more than 10 percent outside the range imposed by the process variable scale (PV_SCALE) parameter. More constrictive setpoint limits can be set using the high (SP_HI_LIM) and low (SP_LO_LIM) setpoint limit parameters. This prevents the setpoint from being set over the full scale (figure 4-38).

By default, the setpoint limits only apply in Auto mode. This restricts the setpoint value that the operator can enter. In other words, the setpoint value does not apply to cascade setpoint input in Cas mode, for example. However, if in the PID block the option "Obey SP limits if Cas or RCas" has been enabled in the control options (CONTROL_OPTS) parameter then the setpoint limits will apply in cascade and remote cascade modes too. Note that the output class blocks do not have the control option and usually are not in Auto mode. The absolute setpoint limits are therefore not really used.

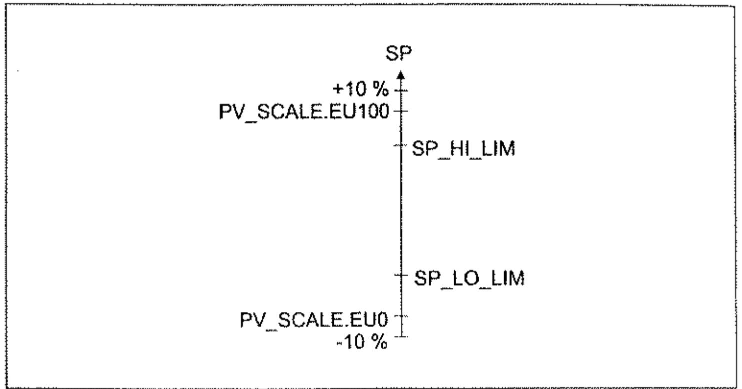


Figure 4-38. The process variable scale and more constrictive limits restrict the setpoint.

In addition to the absolute setpoint limits there are also rate-of-change limits. There are separate parameters for increasing (SP_RATE_UP) and decreasing (SP_RATE_DN) rate of change. The engineering unit for the rate-of-change limits is the process variable unit per second. For example, if the process variable unit is °C the rate-of-change limit unit is °C/s; for % it is %/s. Suppose a new user entry or a change in cascade setpoint input causes a sudden change in setpoint. In that case, the block will use an internal "working setpoint" that slowly ramps to the new setpoint at a rate determined by the rate-of-change limit (figure 4-39).

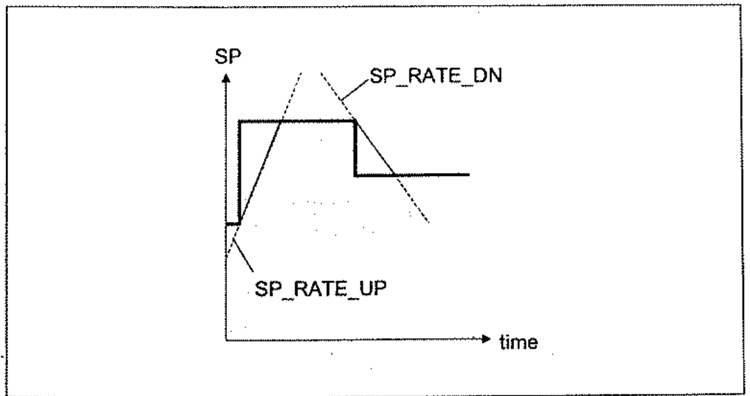


Figure 4-39. Setpoint rate-of-change is restricted within limits.

For example, if the rate of change limit is 5%/s and the setpoint makes a sudden change of 20%, it will take four seconds for the internal "working setpoint" to ramp to this new value.

The rate-of-change limiting can be disabled by setting the limits to zero, in which case the setpoint will be used immediately as is. For control class blocks, rate limiting applies only in Auto mode. For output class blocks, rate limiting applies in Auto, Cas, and RCas modes. Because the PID block has no rate-of-change limit on the output, rate limiting is instead applied on the AO block setpoint.

If the setpoint is limited this will be indicated in the status of the back-calculation output as "Limited High" or "Limited Low." For example, the output of a primary PID in a cascade is used as the setpoint of the secondary PID. If the primary output increases until it touches the high setpoint limit of the secondary, the secondary will set the back-calculation output status as "Limited High." This in turn will prevent the primary PID from increasing its output further; in other words, the primary output will stop at the secondary setpoint limit. However, the primary PID will not go into IMan mode.

A "Constant" limit condition will be set on the output (OUT) in Man, IMan and LO modes or on inputs which are not connected until they are changed by an operator. If the back-calculation input status (BKCAL_IN) is constant the output (OUT) value will follow the back-calculation input. If the input of the PID block is constant the PID block will stop integration.

Mode Shedding and Return

When the operating conditions make the set Target mode impossible, the Actual mode will change. For example, if the cascade setpoint input (CAS_IN) is "Bad," the value is unusable and the mode can no longer be Cas. If the primary input (IN) is "Bad," no control can continue and thus the mode should be Man. Under these and other conditions, the Actual, and optionally the Target, mode will change. Automatic change of mode is called "mode shedding." The Actual mode can only shed to a mode that is enabled in the Permitted mode element of the block mode parameter. When the condition that is causing the shedding disappears, the block will normally return to the previous mode. Mode always sheds to the next higher priority Permitted mode. Out-of-service has the highest priority; remote output has the lowest (see table 4-3). Mode never sheds to remote-cascade or remote-output.

Table 4-3. Mode priorities.

Mode	Description	Priority	Remark
OOS	Out of Service	7	Highest
IMan	Initialization Manual	6	
LO	Local Override	5	
Man	Manual	4	
Auto	Automatic	3	
Cas	Cascade	2	
RCas	Remote Cascade	1	
ROut	Remote Output	0	Lowest

If the process variable input is "Bad," the Actual mode in an input class block will shed to manual. Normally, it will return to the normal operating (Target) mode when the bad condition disappears. By enabling the status option (STATUS_OPTS) "Target to Manual if Bad IN," the Target mode will also be set to manual. As a result, the input block will remain in manual mode even after the bad condition has cleared. The operator must then put it back into operation.

If the cascade setpoint input (CAS_IN) value is "Bad"—for example, because of communication problems—and the Actual mode is Cas then the mode will shed. If the mode is not Cas the condition will be ignored because, in general, the mode is not affected by the status of unused inputs. If the back-calculation input (BKCAL_IN) status is "Bad" (for example, because it is not connected or there are communication problems), the block mode will shed to IMan.

Remote Mode Shedding

The remote modes (RCas and ROut) are not often used, and only a few control class blocks and output class blocks support them. They are present to allow other applications to write the block setpoint or even the block output directly. In other words, the remote output mode does not really use the block functionality at all during normal operation. The remotely written value is passed on directly to the next block. However, should the "link" between the block and the remote application fail the mode of the block will shed to a higher-priority mode that allows the block to take over the functionality. For example, a loop may be controlled using some advanced multiple variable and multiple constraint control software that writes its output to the output parameter of a PID block. However, if the "link" is lost, mode will shed to Auto, and regular PID control will take over. The loop will still operate in a less advanced way, but while maintaining high availability.

The remote cascade setpoint (RCAS_IN) is set in the same unit as the process variable, that is, PV_SCALE. The remote output (ROUT_IN) is set in the same unit as the output.

The remote applications are not connected using regular function block links from output to input parameters. Their setpoint or output is written to parameters that are contained: remote cascade input (RCAS_IN) and remote out input (ROUT_IN), respectively. Contained parameters do not have the loss of communication detection mechanisms that input parameters do. Therefore, all the remote cascade input and remote output input parameters in the device have common update time-out limits that are set in the resource block: the shed remote cascade (SHED_RCAS) and shed remote output (SHED_ROUT), respectively. When the remote setpoint or output has not been updated for the set time the mode will shed.

When the communication has been reestablished the block can return from the shed. The control and output class blocks that support the RCas and ROut modes have a shed and return option parameter (SHED_OPT) that may be used to configure mode shedding and return preferences in order to balance the interests of safety and availability. The shed option parameter can only be set when the block is in OOS mode during online configuration. There is no such restriction when configuring off line. The following shed and return combinations are available:

- Normal shed, normal return
- Normal shed, no return
- Shed to Auto, normal return
- Shed to Auto, no return
- Shed to Manual, normal return
- Shed to Manual, no return
- Shed to Retained target, normal return
- Shed to Retained target, no return

In other words, there are four options for shedding in the event of fault, which can be combined with two options for return upon recovery.

If the shed option "Normal shed" is selected, then in the event of a shed the Actual mode would change to the next lower priority nonremote mode that has been Permitted and that is possible. For

example, if the cascade input (CAS_IN) is not connected, Cas mode is not possible, and therefore the block will shed to Auto and so on. Alternatively, if you use the shed options "Shed to Auto" and "Shed to Manual," the mode can shed directly to automatic or manual mode. If you select the shed option "Shed to Retained target," the mode will shed to either Auto or Cascade, depending on which one was the mode before switching to remote. In the case of "Normal return," the mode will return to the set Target remote mode upon recovery. This provides the greatest availability for the remote application. For safety reasons, it may be better for the mode to remain in the shed mode. Thus, the safer option "No return" changes the Target mode to the nonremote mode that the block has shed to.

Although the remote modes are rarely used, you must always configure the shed option parameter since the default value is invalid and will lead to a block error. Generally, the control strategy templates that are provided with the configuration tool will ensure that all parameters have valid values.

Common Input Class Function Block Characteristics

The input class of function blocks accesses a physical measurement through a hardware channel from a transducer block. The input function block may process the value before it makes it available to other function blocks on its output. The input function blocks do not have setpoint selection.

The input class of function blocks includes the following:

- Analog Input (AI)
- Discrete Input (DI)

Although it is possible to put an input class function block in manual mode, it is usually set in automatic once and for all and not changed.

Input Channel

The channel (CHANNEL) parameter in each input block is used to select the transducer block from which the primary variable is taken. Many transmitters only have a single integral sensor, which means that the channel parameter really only has one option: 1. However, a multivariable device such as a density transmitter may have both a density and process temperature sensor and may therefore require two corresponding AI blocks. The channel parameter is used in each AI block to select whether the block shall have access to the density or the process temperature. Another example of the use of channel parameters is a device that has multiple input terminals. If the sensors for some reason have been cross-wired, it is possible to correct the sensor assignment to the function blocks without having to physically rewire the sensors at the terminals simply by swapping the channel parameter settings (figure 4-40).

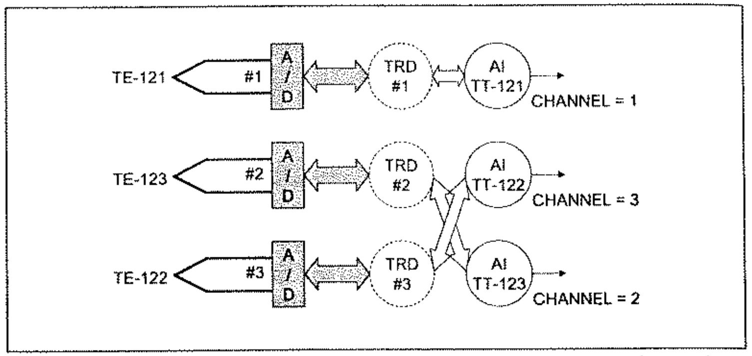


Figure 4-40. The CHANNEL parameter can be used to rewire the inputs in software.

Input Simulation

When enabled, the scalar simulate parameter (SIMULATE) and the discrete simulate parameter (SIMULATE_D) can be used to override the signal from the transducer for system test and troubleshooting purposes. By writing values or status to the Simulate element of the parameter, it is possible to safely test the system response to faults and process conditions that would otherwise be difficult or dangerous to try out. The simulation parameter Transducer element shows the value and status of the signal from the transducer. This parameter is therefore useful for checking whether the block is receiving the correct transducer value. A block alarm will be issued when Simulate is active, and this can also be seen from the block error parameter. Simulation has to be enabled if the device has hardware protection such as a jumper. Lastly, the simulation is activated by the Enable element in the simulate parameter. The status of the simulation protection switch can be seen from the block error parameter in the resource block. The simulation Enable element setting in the function blocks is not retained during a power loss in order to ensure that the device always powers up in the normal operating mode.

Input Status Options

By default, the status on the output parameter of the input block during sensor failure is "Bad - Non Specific." Although this is sufficient to make the mode in a control block shed mode, the reason for doing so will not be visible from a host looking only at the PID block. By enabling the status option "Propagate Fault Forward" the output status will instead be the more specific "Bad - Sensor Failure." This makes it possible to spot the problem from the control block without tracing back to the input block. The status is propagated through any intermediate calculate class blocks. No block alarm will be generated in the input block when this option is enabled. With this primary variable input status, a control class block will generate a block alarm on behalf of the input block. That is, only one of the blocks issues the sensor failure alarm. Therefore, it is a good idea to enable this option in an input block part of a control loop that connects to a control block, but to leave it disabled in an input block that is used for monitoring.

AI - Analog Input Function Block

The analog input block takes the primary value (PRIMARY_VALUE) from the transducer block—for example, pressure, temperature, or flow—and makes it available to other function blocks. It provides a number of the functions expected from a transmitter, such as simulation, linearization, filtering, and alarm (figure 4-41).

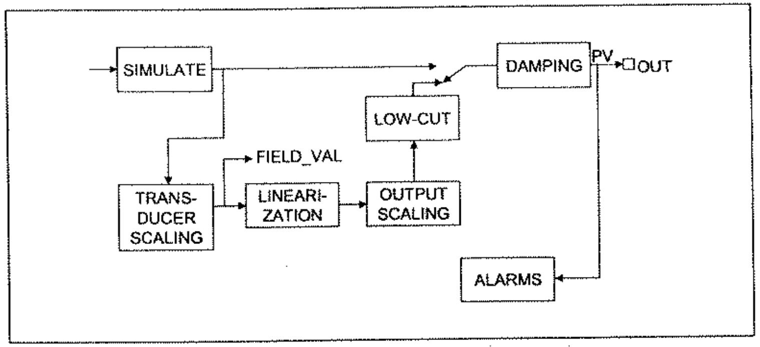


Figure 4-41. AI block schematic diagram.

The AI block implements channel, simulate, damping, and process variable alarms. The transducer scale (XD_SCALE) parameter usually sets the engineering unit that is used by the transducer block to prevent any inconsistencies and subsequent errors. It is important to note that the process variable alarms in the AI block act on the output (OUT) parameter, not on the process variable (PV) parameter. It is therefore important to keep the AI block in automatic mode. This also means that the alarm trip limits must be set within the range specified by the output scale (OUT_SCALE) parameter. Hence, the output scale parameter may have to be set for the sake of the alarms, even though it is not used for scaling. If the AI block is part of a control loop it may be a good idea to detect the alarms in the controller block instead to reduce the number of blocks that have to be monitored.

Damping

For some applications, damping has to be applied to filter out noise on the measurement signal, thus obtaining the process variable (PV) and output (OUT). The process variable filter time (PV_FTIME) parameter is used to set the time constant in seconds of a first-order lag, that is, the time required to reach 63 percent of the steady-state value (figure 4-42). If the time is set to zero seconds, the damping is disabled and the measurement is passed right through.

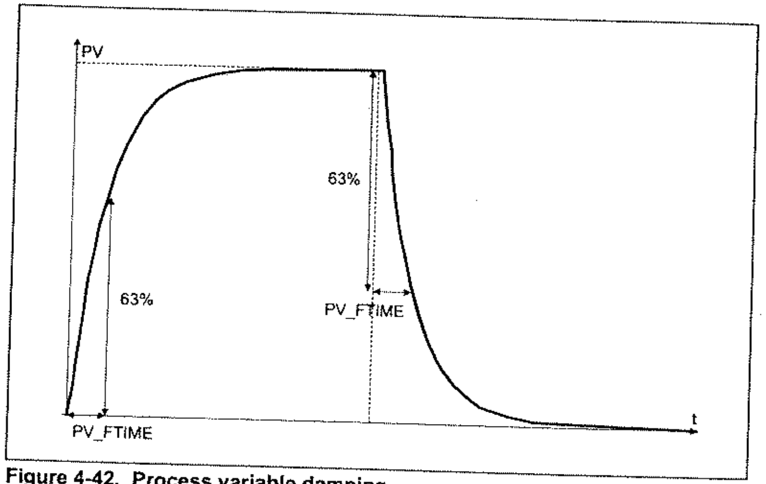


Figure 4-42. Process variable damping.

Linearization

For most measurements, the value from the transducer block is used directly, but for inferred measurements linearization may be applied. The type of linearization that is required is set in the linearization type (L_TYPE) parameter. The following standard linearization options are available:

- Direct
- Indirect, linear
- Indirect, square root extraction

When the "Direct" option is selected, the measured value from the transducer essentially passes through the AI block with no further processing, apart from the damping. No re-scaling of the measurement is performed. The AI block operates over the entire range that the associated sensor is capable of. As a result, there is no need to configure the scaling parameters. In other words, neither XD_SCALE nor OUT_SCALE is applicable. This is the option that is almost always used for temperature, level, flow transmitters, and the like, as well as for pressure transmitters used to measure pressure.

The two "Indirect" options are used for inferred measurements. For the indirect options, the transducer scale (XD_SCALE) and output scale (OUT_SCALE) parameters come into play. The primary value from the transducer block is converted into a percentage of the range that was set in the transducer scale parameter and shown in the field value (FIELD_VAL) parameter. The engineering unit and range of the actual measurement are set in the transducer scale, and the engineering unit and range of the inferred measurement are set in the output scale. The "Indirect, linear" option is used when, for example, a pressure transmitter is used to measure liquid level based on the hydrostatic pressure. For the example in figure 4-5, the XD_SCALE shall be configured 1.49-5.89 kPa and the OUT_SCALE set 0-0.56 m. For level applications, operators very often prefer a reading in percentage terms rather than in an engineering unit. For that purpose, configure OUT_SCALE as 0-100%.

The "Indirect, square root" option is used when a differential pressure transmitter is used for flow measurement. Square root extraction is made on the field value. For example, if for a flow rate of 0-1970 m^3/h an orifice plate produces a differential pressure of 0-1000 mmH2O the XD_SCALE shall be configured 0-1000 mmH2O and OUT_SCALE set 0-1970 m^3/h. The signal characterizer block can be used for linearization using a lookup table. For safety reason, scaling parameters can only be changed in Manual or OOS mode during online configuration. There is no such restriction when configuring offline.

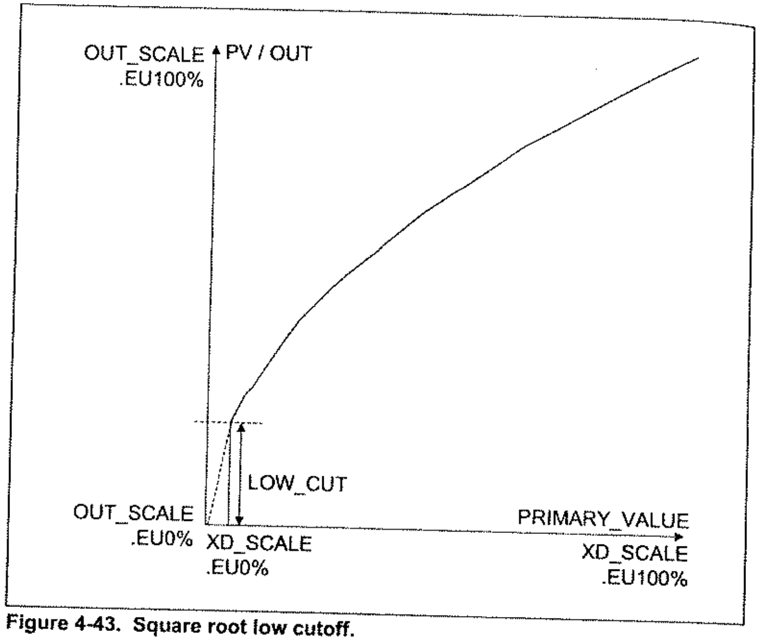


Figure 4-43. Square root low cutoff.

The low cutoff option in the I/O-option parameter (IO_OPTS) is typically used in conjunction with square root extraction to avoid the high loop gain that results from square root extraction at low flows. Cutoff results in a more stable reading at low flows, better control, and the avoidance of "false counts" when totalizing. The cutoff point is set by the low cutoff parameter (LOW_CUT) in the same engineering unit as the output scaling and may be adjusted to suit the application. For flows below this value, the output is zero (0%), for example, for the previous example of differential pressure flow measurement a low flow cutoff of 20% (equivalent to 4% of differential pressure) LOW_CUT corresponds to 394 m3/h. In other words, for flows below 394 m3/h the process variable and output will be 0 m3/h (figure 4-43).

The lower-range value (EU0%) of the scaling parameters can be higher than the upper-range value (EU100%). This is required in some level-measurement applications, and is useful for correcting the reading if a differential pressure transmitter has been mounted the wrong way around.

When you use linearization, it may be a good idea to enable the status option "Uncertain if limited." This provides the option to treat potentially extreme values outside the desired range as "Bad" for safety reasons.

Setting the range should not be confused with Calibration done against a standard, which is performed in the transducer block.

Minimum Configuration

At a minimum, the following parameters must be set:

MODE_BLK, CHANNEL, and L_TYPE.

Common Control Class Function Block Characteristics

The control class function blocks receive inputs from other function blocks and process them, producing output for other function blocks. Most control function blocks have both setpoint and output selection and support the cascade structure.

The control class of function blocks include:

- Manual Loader (ML)
- Bias and Gain (BG)
- Control Selector (CS)
- PD control (PD)
- PID control (PID)
- Ratio (R)
- Setpoint Generator (SPG)
- Output Splitter (OS)

Control class function blocks normally do not pass status down the cascade, for example, a "Bad" input does not become a "Bad" output.

Output Limiting

By default, the output is limited to +/-10 percent outside of the output scale (OUT_SCALE) parameter. For most applications, the output scale will be left at its default value of 0-100%, which represents a fully closed and open valve or a pump stopped or at maximum speed. For this case, the output is thus limited within the range of -10% to +110%. However, for some applications, such as the primary PID in a cascade, the output scale may be set to another range to match the process variable scale in the receiving block. For most control blocks, the output can optionally be limited to a more constrictive high limit (OUT_HI_LIM) and low limit (OUT_LO_LIM). The output limits may be set 10 percent outside of the output scale. These limits may be used to avoid critical conditions. By default, the output limits also apply to Manual mode. However, by enabling the control option (CONTROL_OPTS) "No OUT limits in Manual," the output limits do not apply in this mode. This, of course, means that it is up to the judgment of the operator to enter a safe value.

When the output reaches a limit, this is indicated as "Limited High" or "Limited Low" in the limited condition part of the output status. In Auto, Cas, and RCas modes the limit condition will also be indicated on the back-calculation output (BKCAL_OUT). The limit condition is received on the back-calculation input (BKCAL_IN) of the higher block, for example, from the secondary PID to the primary PID in a cascade control loop. The AO block does not really have an output and therefore no output limits. The limit condition, when received, prevents the upper block from moving its output further in that direction, which thereby prevents reset windup of the integrating action in a PID controller. In other words, the limit condition status information acts as reset windup protection between function blocks that are located in the same or different devices. Note that there is no mode shedding as a result of a limit condition. The PID block has a special back-calculation hysteresis (BKCAL_HYS) parameter to prevent offset caused by the noise rectification of an unstable controller output near the output limit. There are no output rate- of-change limits, but by configuring the setpoint rate of change limits in the receiving block, such as an analog output block, you can avoid sudden process upsets.

Output Tracking (Control Class Local Override)

At times, it is desirable to drive the control output to some form of safety value. In the event of an emergency, it can be operated from some kind of simple local device. The controller could be a backup usually overridden by a main controller, or otherwise you could override the output. Most control class blocks support this function. For output tracking to occur, it must first be enabled from the control options (CONTROL_OPTS) parameter by activating the option "Track Enable." An external tracking value is received on the output tracking value input (TRK_VAL). This value is scaled by the track scale parameter (TRK_SCALE). If the discrete output track input (TRK_IN_D) is true, the control block Actual mode will switch to Local Override, and the scaled tracking value will be converted to the output unit based on the output scale

(OUT_SCALE). However, if the block is in manual mode another control option, "Track in Manual," must also be true in order for the block to shed to local override mode. Once in local override mode, the limit-condition part of the status for the output becomes "Constant," even though the output value is dynamically following the tracking value. The status of the back-calculation, remote-cascade, and remote-output parameters becomes "Not Invited." This ensures that higher control blocks go into IMan mode and do not wind up because of the open loop. The mechanism is the same as for the cascade mode initialization. The discrete tracking input might be an external signal received through a discrete input (DI) block or from an analog alarm block based on an alarm condition, and so on.

By looping the PID output back into the track value input a scheme is achieved in which the PID output is forced to freeze at its last value when the discrete track input is activated, for example, from some external signal.

If the discrete track input goes "Bad," the last usable value is used. If the discrete track input is "Bad" on power-up, it will be set false. If the tracking value goes "Bad," the last usable value will be used. If there is no last usable tracking value, the block output will hold at its present value.

The function blocks in the FOUNDATION Fieldbus programming language already have several safety interlocks built in to deal with sensor failures, and the like. For this reason, the output-tracking feature is seldom used.

Setpoint Limiting (Control Blocks)

The absolute setpoint limits do not apply in Auto mode. However, the setpoint rate-of-change limits apply only in Auto mode. If you set the rate-of-change limit to zero the setpoint will be used immediately, which effectively disables the setpoint rate-of-change limit. When the setpoint is limited this will be indicated in the limit portion of the status element of the setpoint parameter and back-calculation output.

Damping

For some applications, damping has to be applied to filter out noise on the input (IN), thus obtaining the process variable (PV). The process variable filter time (PV_FTIME) parameter is used to set the time constant in seconds of a first-order lag, that is, the time required to reach 63 percent of the steady state, just as for the AI block. If the time is set to zero seconds the damping is disabled and the measurement is passed right through.

Control Options

Some of the control class function blocks make use of one or more options in the control options (CONTROL_OPTS) parameter, which allow the block behavior to be customized. For safety reasons, this parameter is only set when the block is in OOS mode during online configuration, but there is no such restriction when configuring off line. The options that can be set are as follows:

- Enable bypass
- Setpoint tracking in Manual mode
- Setpoint tracking in ROut mode
- Setpoint tracking in LO and IMan modes
- Retain Target SP selection
- Direct Action
- Enable output tracking
- Output tracking in Manual
- Use PV for BKCAL_OUT
- Act on IR
- Use BKCAL_OUT with IN_1
- Obey SP limits if Cas or RCas modes
- No output limits in Manual

The options are best described in the context of their respective features. The "Direct Acting," "Bypass Enable," "SP-PV Track in Man," "SP-PV Track in Rout," and "SP-PV Track in LO or IMan" options are explained for the PID block. The "SP Track retained target," "Obey SP limits if Cas or Rcas," and "No OUT limits in Manual" options are described for the general block characteristics. The "Track Enable" and "Track in Manual" are explained for the control class blocks.

In the past, cascade control setpoint tracking in the secondary PID block was accomplished by setting the primary PID output to the process variable of the secondary controller. In addition to the setpoint tracking options already available in the blocks, you can also achieve tracking in the PID block in the traditional way by configuring the secondary PID back-calculation output to follow the process variable instead of the setpoint. This can be done by using the option "Use PV for BKCAL_OUT." In IMan mode, the primary PID will initialize its output to the secondary process variable. This will make the secondary setpoint identical to the process variable, eliminating error and bumps.

PID - PID Control Function Block

The PID (Proportional, Integral, and Derivative) control block receives a controlled variable on its input (IN) parameter. These variables (for example, pressure, temperature, or flow) are typically received from an AI block, but it's also possible for them to be processed by another block. The PID control block provides a number of the functions expected from a controller, such as filtering, setpoint selection and limiting, the PID algorithm, output selection and limiting, feedforward, setpoint tracking, output tracking, and process variable and deviation alarm (figure 4-44). You can use the PID block to build common control strategies together with other blocks.

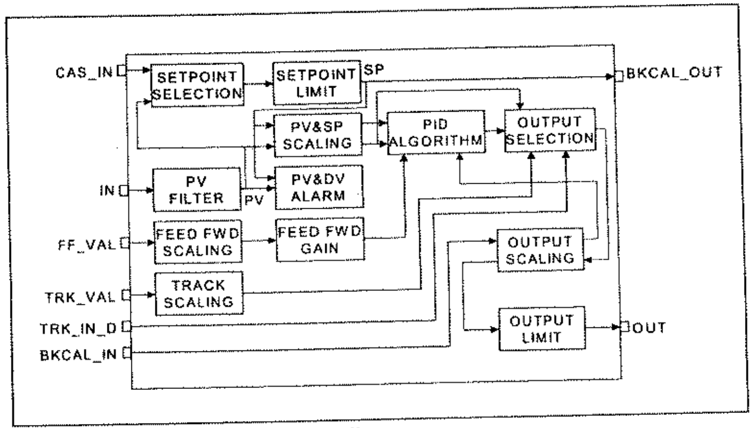


Figure 4-44. PID block schematic diagram.

Tuning

At the heart of the PID function block lies a PID algorithm. To make the tuning constants dimensionless, you must configure a control range in the process variable scale (PV_SCALE) parameter. The engineering unit does not matter; only the range values are used. In other words, for the FOUNDATION Fieldbus programming language the "range" is usually not set in the transmitter (AI block) but rather in the controller (PID block). There are three tuning parameters: Proportional gain (GAIN) parameter (dimensionless), Integral reset time (RESET) parameter in seconds per repetition, and Derivative time constant (RATE) parameter in seconds. You should set reset time to positive infinite (+Inf) to disable integral action. Set derivative time to zero to disable derivative action. It is possible to configure a user interface so it displays the tuning parameters as proportional band and repeats per minute, and so on.

Action

You may configure direct or reverse control action in the control options (CONTROL_OPTS) parameter. By default, the action is reverse, but by enabling the "Direct Acting" option the action instead becomes direct. For direct action, an increase in the process variable results in an increased output. Consequently, for reverse action a decreasing process variable results in an increasing output. The action setting does not affect the deviation alarm.

Action can be set in both the PID and AO block, depending on the type of actuator your system uses. There is risk of inconsistency and confusion because it's unclear if 100 percent means an open or closed valve. For consistency throughout all the loops in the plant, it is a good idea to implement the control strategy such that a PID output of 100 percent always means the valve is open, independent of the actuator being air-to-open or air-to-close. This is achieved by configuring the AO block according to the actuator and the PID block according to the process behavior. The "Increase to close" I/O option in the AO block is used with air-to-close actuators. The "Direct-acting" control option in the PID block is used in processes where increasing process variables necessitates increasing control output, for example, for the chiller valve in a cooling application.

Back-calculation Hysteresis

For some rare cases, a noisy secondary controller output near one of the output limits (OUT_HI_LIM or OUT_LO_LIM) of the back-calculation limit status would cause that limit status to be set during a peak, only to soon return to normal. That is, the secondary controller output will move away from the violated limit in reaction to the primary controller's response of not bringing its output further in that direction when the secondary reaches the limit. If the back-calculation limit status of the secondary were to immediately return to normal, the primary might move its output making the limit condition occur again, causing a "chattering" output. An offset would be created that

cannot be integrated away. The details of this phenomenon involve control theory and are beyond the scope of this book.

However, the back-calculation hysteresis parameter is available to alleviate the problem. When the limited status on the secondary back-calculation output has been set, it will only clear when the output is below the upper limit minus the back-calculation hysteresis or above the lower limit plus the back-calculation hysteresis for high and low output limit, respectively. Since the limit status is not released at once, the primary does not start integration immediately. This minimizes the problem. For most applications, you can leave the back-calculation hysteresis at the default value of 0.5 percent. If the process is noisy, you can set the back-calculation hysteresis in a secondary cascaded controller to a higher value, and it is set in terms of percentage of the output scale.

Setpoint Tracking

Setpoint tracking should not be confused with the retention of the setpoint source selection. Setpoint tracking means that the setpoint value (SP) tracks the process variable value (PV) in selected "manual-like" modes. In a control class block, the "manual-like" modes are these:

- IMan
- LO
- Man
- ROut

By enabling the different options in the control options parameter (CONTROL_OPTS) you can make the setpoint track the process variable whenever the block is in the respective mode. The control options for setpoint tracking are "SP-PV Track in Man," "SP-PV Track in Rout," and "SP-PV Track in LO or IMan."

When switching from a "manual-like" mode like to an "automatic-like" mode there is no proportional bump even in the presence of error. By default, the output bumplessly takes off integrating from the last output value before mode transfer. If no output change at all is desired, the error also has to be eliminated to ensure that there will be no integral action. Setpoint tracking accomplishes this by letting the setpoint track the PV in the "manual-like" modes. Once the mode that causes setpoint tracking is left, the setpoint will be frozen at the last value. Of course, the output will only remain constant as long as the process variable does not change or the setpoint is not changed.

It is a good idea to enable setpoint tracking in the secondary PID block of a cascade to ensure a smooth return to normal operation after the cascade has been broken.

Feedforward

In feedforward control, a disturbance or load is measured and used to calculate the manipulated variable in order to cancel the influence of the disturbance or load on the process. The most common application for feedforward is calculating a material or energy balance on a setpoint to a secondary flow controller in a cascade loop. The feedforward contribution in the PID is added to the output. It should therefore be done in the primary controller to manipulate the setpoint of the secondary controller in the cascade. The feedforward signal is received on the feedforward value (FF_VAL) parameter, which may be input from any function block but is typically input from an AI block (figure 4-46). The contribution of the feedforward signal is controlled using the feedforward gain (FF_GAIN) parameter. To make the feedforward gain dimensionless the feedforward value is first scaled to percentage terms using the (FF_SCALE). The feedforward gain multiplies the scaled feedforward value before it is added to the PID algorithm result to become the output (OUT).

The feedforward signal is taken into consideration when the output is limited. This eliminates the risk that the output gets too large because of the feedforward signal. Likewise, the feedforward signal is taken into consideration during initialization of the output in IMan mode. This ensures that no external calculations are required to make mode transfers bumpless.

A classic application is three-element, drum-level control for a boiler. An increase in load and steam flow adjusts the setpoint in the secondary PID controller block. This results in an increase in feedwater flow before the drum level is affected (figure 4-45).

Output Scaling for Cascade Primary

By default, the output scaling is 0 to 100 percent. The output scaling makes it possible to assign an engineering unit range to the PID block output. The range is determined by the output scale parameter (OUT_SCALE). If the block operates as a primary PID in a cascade control application the output of the block is then the setpoint of another. It is then possible to scale that setpoint (i.e., this output) to match the process variable scale of the secondary PID, for example, in units of flow.

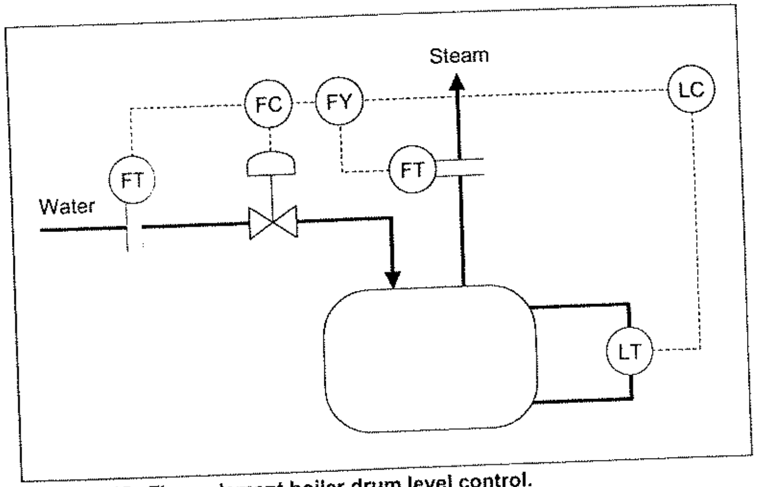


Figure 4-45. Three element boiler drum level control.

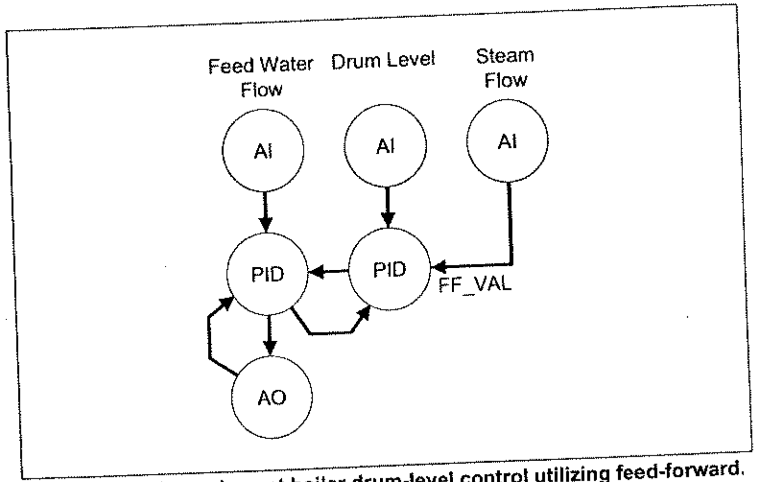


Figure 4-46. Three-element boiler drum-level control utilizing feed-forward.

For most applications, the output scale (OUT_SCALE) parameter is set from 0 to 100 percent. However, for the primary PID in a cascade the output scale is usually set in engineering units, typically to the same range as the controlled variable of the secondary PID (figure 4-48). In the example in figure 4-47, the output scale of the level controller is set in the same range as the process variable scale of the flow controller.

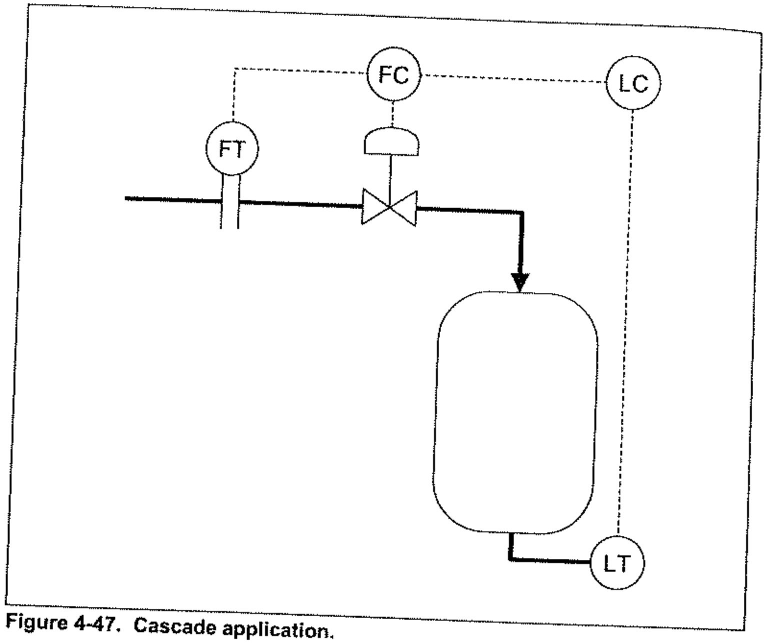


Figure 4-47. Cascade application.

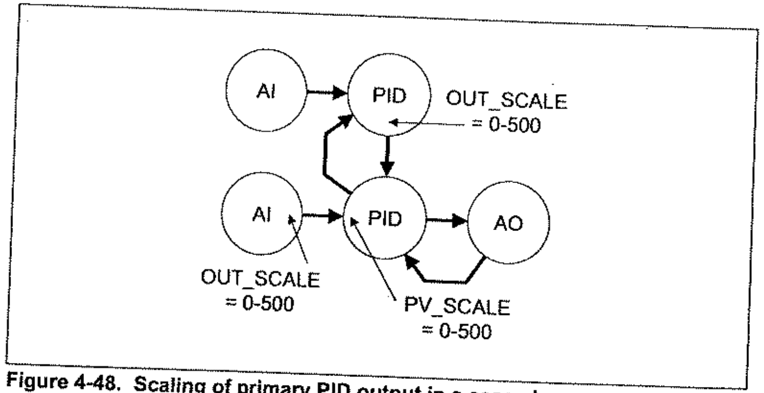


Figure 4-48. Scaling of primary PID output in a cascade.

Bypass

When the controlled variable input (IN) to a secondary PID block in a cascade is "Bad," the PID algorithm in the secondary PID cannot function. However, it may be bypassed, which lets the primary PID control the analog output block directly by simply going through the secondary PID. Not all control schemes are stable in bypass, so you must enable the bypass feature by setting the option "Bypass Enable" in the control options (CONTROL_OPTS) parameter. Bypass is activated with the bypass (BYPASS) parameter. By default, the bypass function can only be activated in manual or OOS mode during online configuration, but there is no such restriction when configuring off line. However, if you enable the feature selection (FEATURE_SEL) option "Change of bypass in Auto allowed" in the resource block the bypass function can also be activated in automatic mode. Thus, assuming that the control option has been enabled in advance, the procedures for activating and deactivating the bypass function in the secondary PID are as follows:

Activate bypass—

1. Set Target block mode to manual
2. Set bypass On (BYPASS)
3. Set Target block mode to cascade

Deactivate bypass—

1. Set Target block mode to manual
2. Set bypass Off (BYPASS)
3. Set Target block mode to cascade

When bypass is active the setpoint is passed to the output, bypassing the PID algorithm. The process variable and output scaling are taken into account, as is the feedforward value, if it is used. To ensure bumpless transfer into bypass, the output (of the secondary) is passed to the primary PID block over the back-calculation path. This initializes the output of the primary PID when bypass is activated using the standard cascade mode initialization mechanism. To ensure bumpless transfer out of bypass, the process variable (of the secondary PID) is passed to the primary PID over the back-calculation path. This initializes the output of the primary PID when bypass is deactivated. Since on deactivation the process variable and setpoint is equal, deviation is zero, so there will be no bump on the output.

Unless the process precludes it, it may be a good idea to enable the bypass control option in secondary PIDs. This will make it easy to activate *bypass should the transmitter associated with the secondary PID have to be removed for maintenance or calibration.*

Minimum Configuration

As a minimum, the following parameters must be set:

MODE_BLK, BYPASS, and SHED_OPT.

Common Calculate Class Function Block Characteristics

The calculation class of function blocks receives inputs from other function blocks and processes them, which produces output for other function blocks. The calculate class of function blocks supports output selection but not the cascade structure. Typically, these blocks are always in automatic mode.

The calculation class of function blocks includes the following:

- Signal Characterizer (SC)
- Lead/Lag (LL)
- Integrator (IT)
- Input Selector (IS)
- Arithmetic (AR)
- Timer (TMR)
- Analog Alarm (AAL)

Even if the block is in manual, the calculated result can still be seen from the pre-output (PRE_OUT) parameter. Once the mode is switched back to automatic, the output (OUT) will ramp to the pre-output value within the balance time (BAL_TIME). This will prevent the process from being bumped.

AR - Arithmetic Function Block

The arithmetic block provides a general-purpose calculation capability. It may be used to perform common computations used in the process industry, such as pressure and temperature compensation of flow, hydrostatic tank gauging, and controlled flow setpoint in ratio control (figure 4-49).

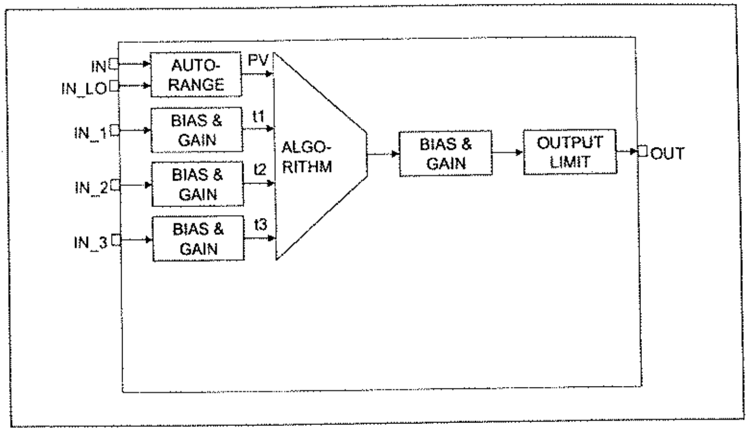


Figure 4-49. Arithmetic block schematic diagram.

The process variable (PV) of the block is obtained from two inputs, the main input (IN) and the low input (IN_LO). For most applications, only the input (IN) is used to obtain the process variable. However, to achieve higher rangeability the low input (IN_LO) can be used for differential pressure flow measurement utilizing two transmitters in an auto-ranging scheme (figure 4-50).

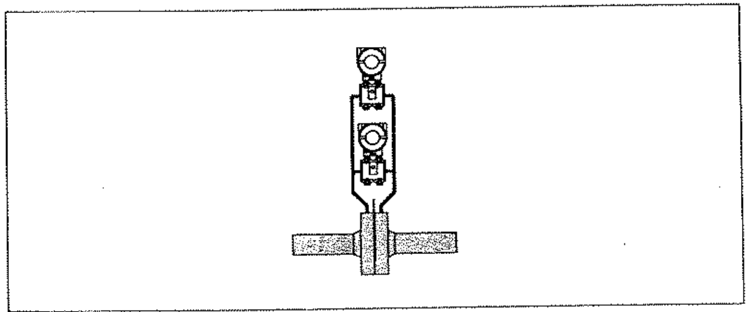


Figure 4-50. Stacked pressure transmitters for greater flow rangeability.

It is quite possible that the two input readings may differ slightly even though they measure the same differential pressure. A single switchover point would therefore cause the process variable to jump from one value to the other. Therefore, the value is gradually interpolated from a set low-range point (RANGE_LO) to a high-range point (RANGE_HI). The range points should be configured for values below but near the upper range of the lower input value. When the input (IN) is within these values, the process variable is interpolated between the input and low input (figure 4-51). The status of the low input is used for the lower half of the range, and the status of the (high) input is used for the higher half of the range. If the low input is not used, both range points should be configured with values lower than the lowest expected value for the input (IN).

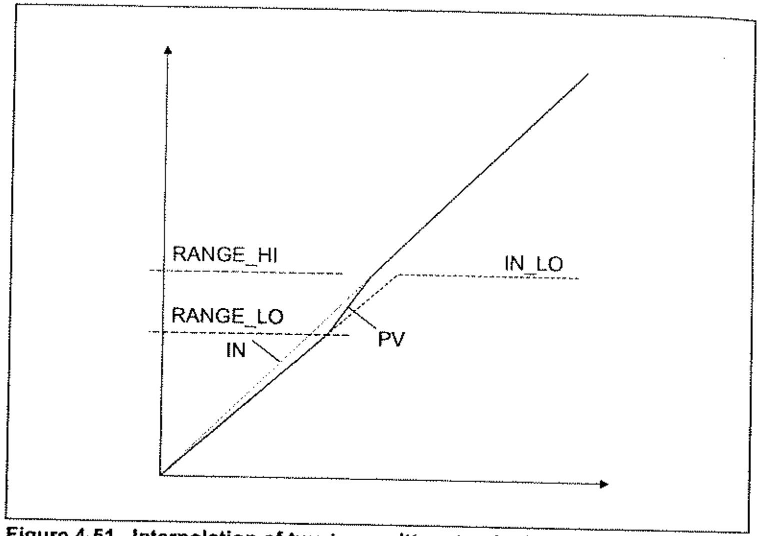


Figure 4-51. Interpolation of two transmitters to obtain PV.

There are three additional inputs: input one (IN_1), input two (IN_2), and input three (IN_3). Each of the three inputs are first added to a bias (BIAS_IN_1, BIAS_IN_2, and BIAS_IN_3) and are then multiplied by a gain (GAIN_IN_1, GAIN_IN_2, and GAIN_IN_3). This enables units to be converted and other adjustments to be made to the input before it is used in the main algorithm. For example, by setting a gain to -1 the input will be subtracted instead of added. The results of the applied gain and bias are not available as visible parameters. However, for reasons of simplicity they are illustrated as "tn" in the block diagram and in the flow compensation equations given later in this section. It is possible to decide whether "Uncertain" or "Bad" inputs should be used by configuring the input options (INPUT_OPTS) parameter accordingly. The available options are as follows:

- IN Use uncertain
- IN_LO Use uncertain
- IN_1 Use uncertain
- IN_1 Use bad
- IN_2 Use uncertain

- IN_2 Use bad
- IN_3 Use uncertain
- IN_3 Use bad

By enabling any of the "Use uncertain" options the respective input value will be used as if "Good" even if the status is "Uncertain." Similarly, the option "Use bad" makes it possible for the input value to be used as if it was "Good" even though the status is "Bad."

For maximum versatility, the block supports several algorithms that the user may choose from using the arithmetic type (ARITH_TYPE) parameter. Each option has a name that indicates its typical application, but each can of course be used with other calculations as well. Depending on the input, some equations may result in division by zero or the square root of a negative number or other illegal operations, or they may result in extreme values. To prevent control from being upset, in most cases the output is first limited by compensation limits and always limited by output limits. The result of square root of a negative value becomes the root of the absolute value with a negative sign. The arithmetic functions available are as follows:

- Flow compensation, linear (limited)
- Flow compensation, square root (limited)
- Flow compensation, approximate (limited)
- BTU flow (limited)
- Traditional Multiply Divide (limited)
- Average
- Traditional Summer
- Fourth-order polynomial
- Simple HTG compensated level

Flow Compensation, Linear

$$OUT = PV \cdot \left[\frac{t1}{t2}\right] \cdot GAIN + BIAS$$

Example:

$$Q_b = Q_f \cdot \left[\frac{P}{T}\right] \cdot K$$

Gas flow compensation for linear transmitters, for example, turbines.

Flow Compensation, Square Root

$$OUT = PV \cdot \left[\sqrt{\frac{t1}{t2 \cdot t3}}\right] \cdot GAIN + BIAS$$

Example:

$$Q_b = Q_f \cdot \left[\sqrt{\frac{P}{T \cdot Z}}\right] \cdot K$$

Gas flow compensation for differential pressure producers, for example, orifice meters.

Flow Compensation, Approximate

$$OUT = PV \cdot \left[\sqrt{t1 \cdot t2 \cdot t3^2}\right] \cdot GAIN + BIAS$$

Examples:

$$Q = L \cdot [\sqrt{L}] \cdot K$$

Square root of third power for open-channel flow measurement using square-notch weir of Parshall flume.

$$Q = L \cdot \left[\sqrt{L \cdot L^2}\right] \cdot K$$

Square root of fifth power for open-channel flow measurement using V-notch weir.

BTU flow

$$OUT = PV \cdot [t1 - t2] \cdot GAIN + BIAS$$

Example:

$$Q_{heat} = Q_{vol} \cdot [t_1 - t_2] \cdot K$$

Heat flow computation

Traditional Multiply Divide

$$OUT = PV \cdot \left[\frac{t1}{t2} + t3\right] \cdot GAIN + BIAS$$

Example:

Ratio control is a form of control that is used when blending an additive to a main component by keeping the ratio of the additive flow (controlled flow - Qc) to the main component flow (wild flow - Qw) constant (figure 4-52).

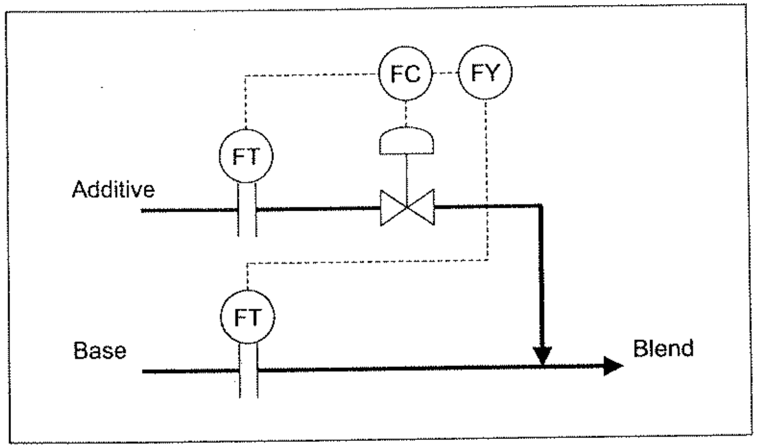


Figure 4-52. Ratio control.

There are two ways of performing this control, either multiplying or dividing. In either case, the AR block is used to calculate the desired additive flow, which is then used as a setpoint in the PID block.

$$Q_{SP} = Q_{wild} \cdot Ratio$$

The result of multiplying a ratio is used as cascade setpoint input for PID block.

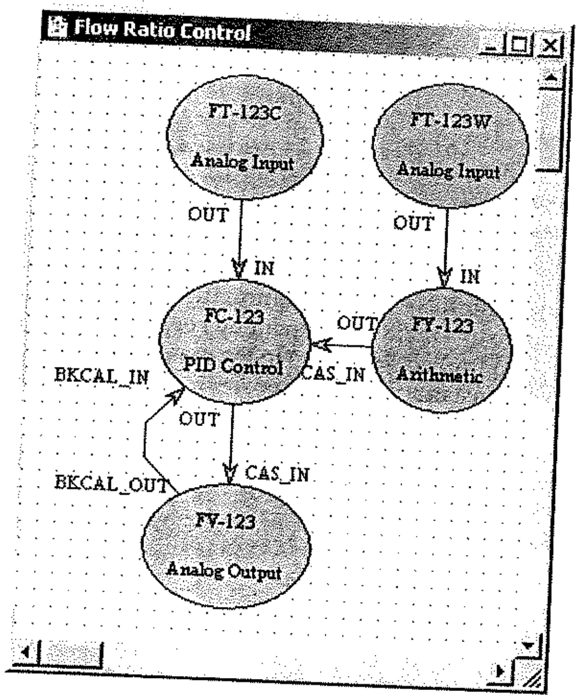


Figure 4-53. Ratio control strategy.

Using the "traditional multiply-divide" algorithm in the arithmetic block you may implement either scheme of ratio (figure 4-53). In the multiplying method, a wild base flow of 3600 kg/h and a ratio of 0.5 would give a controlled additive flow setpoint of 1800 kg/h. If done using the division method, the corresponding ratio would be 2 to achieve the same controlled flow setpoint. In both cases, it is important to remember to set the compensation limit values such that the factor multiplying the process variable is not limited within the normal operating range.

Average

$$OUT = \frac{PV + t1 + t2 + t3}{number\_of\_inputs\_used} \cdot GAIN + BIAS$$

Example:

$$t_a = \frac{t_1 + t_2 + t_3 + t_4}{number\_of\_working\_sensors}$$

Output becomes average of four temperature inputs.

Traditional Summer

$$OUT = (PV + t1 + t2 + t3) \cdot GAIN + BIAS$$

Fourth-order Polynomial

$$OUT = (PV + t1^2 + t2^3 + t3^4) \cdot GAIN + BIAS$$

HTG Compensated Level

$$OUT = \frac{PV - t1}{PV - t2} \cdot GAIN + BIAS$$

Example:

When a single pressure transmitter is used to measure liquid level the indication varies depending on product density because the pressure transmitter really measures mass. To correct the level, a second transmitter is used to compensate for the density. For closed tanks, a third transmitter is added at the top to compensate for the pressure of the vapor in the empty space above the liquid. Figure 4-54 shows a typical arrangement.

The following formula computes tank level compensated for changes in liquid density and ullage pressure:

$$L_c = \frac{P_b - P_t}{P_b - P_m} \cdot h_{bm}$$

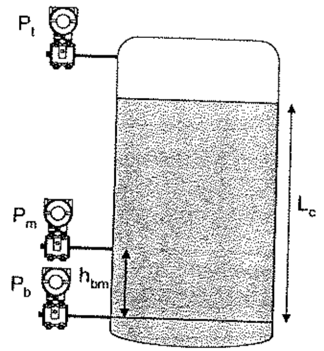


Figure 4-54. Hydrostatic pressure level measurement.

The bottom transmitter connects to the main input, the top transmitter to input 1, and the middle transmitter to input 2. The output gain should be configured to the height between the bottom and middle pressure transmitter.

Limiting

The part of the equation within square brackets is limited by the low-compensation limit (COMP_LO_LIM) parameter and a high-compensation limit (COMP_HI_LIM) parameter, which restrict the amount of correction applied on the process variable. Lastly, to enable output unit conversion and other adjustments to the output, there is also a gain (GAIN) and bias (BIAS) for the output. The block output (OUT) is restricted within a high limit (OUT_HI_LIM) and a low limit (OUT_LO_LIM). The process variable unit (PV_UNITS) and output unit (OUT_UNITS) parameters do not affect the function of the block; they are there for information purposes only.

Minimum Configuration

As a minimum, the MODE_BLK parameter must be set. Remember to set the bias and gain for inputs and output. Left at their default values, inputs may not be taken into account or division by zero may result. For the algorithms in which compensation limiting is employed, you must set the limits so they do not interfere. Similarly, you must set the range for the low input auto- ranging.

Common Output Class Function Block Characteristics

The output class of function blocks receives setpoint input from other function blocks and may process the value before passing it to physical output over a hardware channel through a transducer block. The setpoint is the most important parameter in the output class blocks. Output class function blocks have setpoint selection and possibly output selection too, although output selection is not really used.

The output class of function blocks includes the following:

- Analog Output (AO)
- Discrete Output (DO)

In an output class block, the local override mode can have two functions. On the one hand, it can mean that the fault-state shutdown function in the block is active. On the other, it can mean that the output device has some form of local override switch that has been activated. For example, it is common for electrical valve actuators to have a hand/auto lever that disengages the motor and instead engages a handwheel. During hand operation, the mode would be local override. Although it is possible to put an output class function block in automatic mode, the Target is usually set in cascade once and for all and not changed. The mode would be Auto during local operation but Cas for remote operation. The output function blocks do not really use the output selection. The mode of the control loop is usually set by the block mode parameter in a control class block, for example, the PID block.

Output Channel

The channel (CHANNEL) parameter in each output block is used to select the transducer block to which the output goes, just as for the input class blocks. Most output devices only have a single output. In this case, the channel parameter really only has one option: 1. However, a multivariable device such as a conventional signal converter may have many options. For example, if the output terminals for some reason have been cross-wired in a device that has multiple outputs, it is possible to correct the output terminal assignment to the function blocks without having to physically rewire the outputs at the terminals. This is done simply by swapping the channel parameter settings.

Output Options

Most of the output class function blocks make use of one or more options in the I/O-options (IO_OPTS) parameter that make it possible to customize the block behavior. For safety reasons, this parameter is only set when the block is in OOS mode during online configuration. However, there is no such restriction when configuring off line. The options that can be set are these:

- Invert
- Setpoint tracking in Manual mode
- Setpoint tracking in LO mode
- SP Track retained target
- Increase to close
- Fault State to value
- Use Fault State value on restart
- Target to Man if Fault State activated
- Use PV for BKCAL_OUT

The options are best understood in the context of their respective features. The "SP Track retained target" option is described in this chapter for the general block characteristics. The "Use PV for BKCAL_OUT" and "Increase to close" options are explained in this chapter for the AO block. The "Fault State to value," "Use Fault State value on restart," and "Target to Man if Fault State activated" are discussed in this chapter on the common output class function block characteristics.

Fault State Shutdown (Output Class Local Override)

When devices or communication fail their associated controls cannot continue. Most often, you should shut down the affected loop or at least not attempt control. This function is called "fault state." Diagnostics performed all the way from the sensor are responsible for detecting faults. Ultimately, however, it is the output class blocks that go into local override mode to shut the loop down. When the fault state is activated, the output class block goes into local override mode. For the fault-state function to work, it must be enabled using the "Fault State Supported" option in the feature selection (FEATURE_SEL) parameter in the resource block. Since this feature increases the safety, it is a good idea to make use of it. The fault-state function can be activated in four basic ways:

- "Initiate Fault State" status on the cascade setpoint input

- The cascade setpoint input communication watchdog times out
- Power-on
- Forced from the resource block

There is a status option in most function blocks that sets their output status as "Initiate Fault State" if a critical input parameter is "Bad," usually because of some error or failure. When this status is received on the cascade setpoint input (CAS_IN or CAS_IN_D) parameter of an output class block it will go into LO mode. In the event the period of time in which the communication of the cascade input parameter is lost is longer than the fail-safe time (FSTATE_TIME) parameter, the block will also go into LO mode. The fail-safe time parameter is set in seconds (figure 4-55). The shorter the time set, the safer the loop becomes. However, too short a time may result in annoying spurious trips that reduce the availability of the loop. Two seconds is a typical value for time-out that strikes a good balance between safety and availability. By enabling the I/O-option "Use Fault State value on restart" the fail-safe value is used as a known "power-up" initial output value when the device is switched on. Additionally, there is a fault-state set parameter in the resource block that may be used to instantly force all output class function blocks into their preconfigured Fault State, ignoring the fault state time.

By default, the setpoint will freeze at its last value when Fault State is activated. This allows the process to continue in its last position but without any further manipulation of the output. This option provides availability since the process is not interrupted. Depending on the process and the actuator equipment that you have, a safer option may be to shut the loop down by bringing it to a known safe value. If the I/O-option "Fault State to value" is enabled, the fault-state value (FSTATE_VAL parameter in case of analog, FSTATE_VAL_D in case of discrete) will become the setpoint of the output block. Typically, the fail-safe value is something that means "closed" or "off." The "freeze last value" option may provide better availability because the process continues to run and control can more easily be picked up from the point where it was stopped. The fault-state value in the AO block is, of course, configured in the same units as the setpoint, that is, according to the PV_SCALE.

Once whatever condition caused the fault state disappears the output block will return to the set Target mode, usually cascade, thus

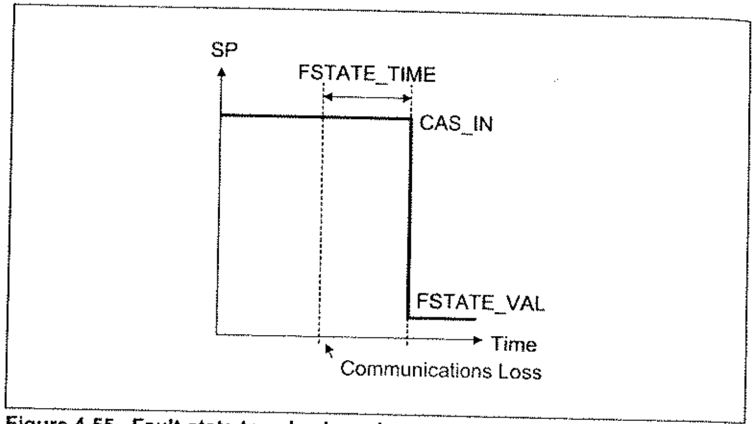


Figure 4-55. Fault-state to value based on communication failure.

allowing control to resume. For some applications, you may prefer that the operator be required to intervene to bring control back to normal operation. For this purpose, you may enable the I/O-option "Target to Man if Fault State activated." In this case, the loop remains shut down even after the fault clears. Since the operator usually only accesses the loop for operation from the PID block you may prefer to use the similar status option in the PID block whenever the application allows.

Most of the time, the control class block is located in the same device as the output block. For example, the PID is usually allocated to the valve positioner. Therefore, there is no communication loss possible for the cascade setpoint input of the AO block. Instead, what can fail and make the PID input "Bad" is the communication between the input block in a transmitter and the PID in the positioner. For this reason, it is a good idea to enable the "IFS if BAD IN" status options in the PID block.

WARNING—Ultimately, it is the hardware that has to ensure that the fault-state setting is acted upon. In the presence of certain faults, such as power loss (electrical, pneumatic, hydraulic, etc.), the fault state cannot take effect. Ultimately, the hardware must also be implemented to fail safely in its deenergized state by using traditional methods such as spring return. A safety assessment of each loop will tell you whether the shutdown functionality provided by the FOUNDATION Fieldbus is sufficient or if additional SIL-rated equipment is required.

Readback

The actual status of the output hardware can be seen at the readback parameter (READBACK in case of analog, READBACK_D in case of discrete). This makes it possible to see, for example, the actual position of a control valve or the actual state of an on/off valve. The readback value is taken backward through the transducer scale (or state, in the case of discrete), action type, and process variable scaling (or state, in the case of discrete) to obtain the process variable (PV in case of analog, PV_D in case of discrete). There is an option to use this actual value for cascade initialization to provide true bumpless transfer and reset windup protection.

Readback Simulation

You can use the scalar simulate parameter (SIMULATE) and the discrete simulate parameter (SIMULATE_D), when they are enabled, to override the readback signal from the transducer for system testing and troubleshooting purposes. By writing values or status to the Simulate element of the parameter, it is possible to safely test the system response to faults that would otherwise be difficult or dangerous to try out. The simulation parameter Transducer element shows the value and status of the signal from the transducer. This parameter is therefore useful for checking whether the block is receiving the correct transducer value and the status of the readback actuator value from the transducer. A block alarm will be issued when Simulate is active, and it can also be seen from the block error parameter. Simulation has to be enabled if the device has hardware protection such as a jumper. Lastly, the simulation is activated by the Enable element in the simulate parameter. The status of the simulation protection switch can be seen from the block error parameter in the resource block. The simulation Enable element setting in the function blocks is not retained during a power loss. This ensures that the device always powers up in the normal operating mode. The simulation parameter in the output class blocks does not write to the transducer block. That is, if the output device, for example, a valve, has to be operated by hand the correct way to go about it is to set the block mode to Automatic and write the setpoint.

Output Status Options

During an output failure, the status on the back-calculation output of an output block is, by default, just "Bad - Non Specific." Although this is sufficient to make the mode in the higher control block shed, the reason why will not be visible to a host that is simply looking at the PID block. Instead, by enabling the status option "Propagate Fault Backward" the back-calculation output status will be the more specific "Bad - Device failure." This makes it possible to spot the problem from the control block without tracing down to the output block. No block alarm will be generated in the output block when this option is enabled. With this back-calculation input status, a control class block will generate a block alarm on behalf of the output block. That is, only one of the blocks issues the output failure alarm. Likewise, the output blocks do not normally propagate the fact that the fail-safe function or local override mode (operation by hand) is active back to the upstream blocks.

AO - Analog Output

The analog output block receives an input from another function block and may process the value before passing the signal through a hardware channel to a transducer block. This may be, for example, the demanded valve position from a control class block. It provides a number of functions that are expected from a positioner, such as limiting, signal reversing, actual position feedback, simulation, and fail-safe function.

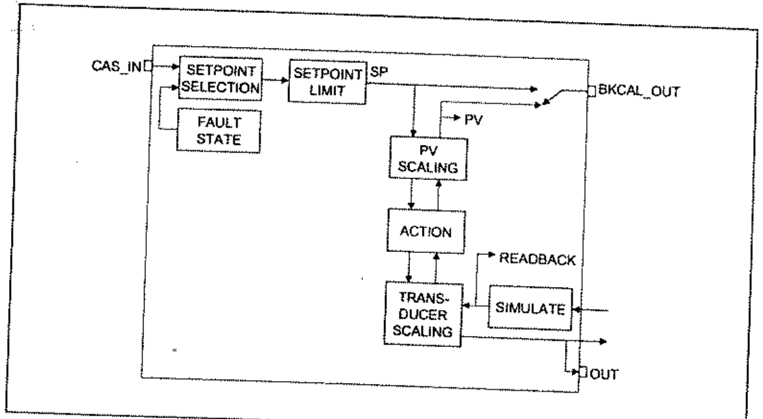


Figure 4-56. AO block schematic diagram.

The desired output value is normally received on the cascade input (CAS_IN), typically from a PID control block. Therefore, the AO block is typically set in Cas mode once and for all and then left there. The demanded valve position, conveyor speed, or pump rate thus becomes the setpoint, which is the most important parameter in the block.

When the fault state is active, the fault-state value becomes the setpoint and will also be seen on the back-calculation output (BKCAL_OUT). This makes the upstream block initialize its output using the standard cascade initialization mechanism so as to provide bumpless transfer out of the fault state.

Setpoint Limiting (Output Blocks)

The absolute setpoint limits do not apply in Auto mode. The setpoint rate-of-change limits apply only in Auto, Cas, and RCas modes. By setting the rate-of-change limit to zero the setpoint will be used immediately, which effectively disables the setpoint rate-of-change limit.

Action

The process variable scale (PV_SCALE) and transducer scale (XD_SCALE) parameters are used to transform the setpoint to a value that is accepted by the transducer block. The engineering unit for the process variable scale does not affect the block operation. The transducer scale unit usually sets the engineering unit of the associated transducer block to eliminate any inconsistencies and subsequent block errors. Control valve positioners normally operate in 0 to100 percent, which represents the closed and open endpoints of the valve travel. However, converters for current or pneumatic signal output need to have their transducer scale set to the appropriate operating range and unit. For example, in a fieldbus-to-pneumatic signal converter the PV_SCALE may be configured 0-100 percent and XD_SCALE set to 3-15 psi.

You may configure an increase- or decrease-to-close action in the I/O-options (IO_OPT) parameter. By default, the action is decrease-to-close and is typically used with air-to-close (fail- open) actuators. However, by enabling the "Increase-to-close" option it instead becomes increase-to-close for use with air-to-open (fail closed) actuators. For increase-to-close action, an increase in the setpoint (PID output) results in a closing valve. Consequently, for decrease-to-close action a decreasing process variable results in a closing valve.

Action can be set in both the PID and AO block and depends on the type of actuator used. There is a risk of inconsistencies and confusion if 100 percent setpoint sometimes means open and other times means a closed valve. For consistency throughout all the loops in the plant, it is a good idea to implement the control strategy such that a PID output of 100 *percent always means the valve is open—independent of the actuator being air-to-open or air-to-close. This is achieved by configuring the AO block according to the actuator and configuring the PID block according to the process behavior. The "Increase-to-close" I/O-option in the AO block is used with air-to-close actuators. The "Direct-acting" control option in the PID block is used in processes where an increasing process variable requires increasing control output, for example, for the chiller valve in a cooling application.*

The output (OUT) parameter is made available to show a copy of the value that the AO block passes to the transducer block over the hardware channel. The output parameter is displayed in the same unit as the readback value and the same unit as used in the transducer block, that is, as set in the transducer scale.

True Windup Protection and Bumpless Transfer

You can see the actual valve position or true reflection of other types of final control element output from the readback parameter in the transducer scale units. The actual output value is converted into a process variable (PV) by doing the transducer and process variable scaling in reverse while taking the action into account. The back-calculation output is normally a copy of the selected setpoint, but by enabling the I/O-option "Use PV for BKCAL_OUT" the actual output, for example, the true valve position, is instead used for the back-calculation output for the normal cascade initialization. This means that if the cascade has been broken it can be returned to normal operation without jerking the actual valve position. In other words, it could achieve a truly bumpless mode transfer in the same way as an additional feedback position transmitter achieved it in the past but without the additional instruments and wiring. To benefit from this feature it is a good idea to implement the entire control loop using FOUNDATION Fieldbus function blocks.

The actual valve position is not available as a block output from which a link can be made. However, it is not really required. The user does not have to configure any interlocks with the PID to achieve true bumpless transfer based on actual valve position because such transfer is a standard function built into the blocks.

If the control valve reaches either its fully opened or fully closed endpoint the AO block will set the corresponding limit in the status element of the back-calculation output parameter. This tells the PID block to not push its output further in that direction. This action prevents reset windup in the PID block in the same way that feedback from limit switches did in the past.

The operator normally sets the loop mode from the block mode parameter in the PID block. To stop control, the mode in the PID block would be set to Man. If the output device for some reason has to be operated by hand, the best way to hand-operate the analog output block is to put the mode to Auto and enter the setpoint.

Minimum Configuration

As a minimum, the MODE_BLK, CHANNEL, and SHED_OPT parameters must be set.

Schedule

Every function block in the control strategy is executed repeatedly several times per second. Each time the block output is passed to the input of the next block. The period during which the function blocks are being executed is called a macro cycle. External function block links are communicated on the network, whereas internal links are passed to the next block within the device. External links are communicated after the originating block has been executed but before the receiving block is executed. This ensures that all blocks are using values with a minimum of delay. The function block link communication is called "operational traffic." To achieve the correct order of execution and communication, the configuration tool synchronizes in a schedule the execution of the function blocks in the different devices on the network and the communication of the external links.

Scheduling is the most common way to coordinate function block execution. It is selected using the cycle type selection (CYCLE_SEL) parameter in the resource block. There is one schedule for each H1 network in the system. It takes into account the execution time of every function block and the time required for communicating each external link, and it ensures that the blocks are executed according to the signal flow through the strategy. The schedule is automatically generated by the configuration tool and downloaded to the devices as part of the control strategy. The schedule tells the device when to execute its function blocks and tells the LAS how to synchronize the communication of the external links. Every device has its own microprocessor, which together form a multitasking multiprocessor system in which several function blocks are being executed at the same time in the various devices according to the schedule. The macro cycle is usually not longer than the time it takes to communicate the external links in the strategy plus background traffic. However, if a device has exceptionally long block execution time the loop execution will be slow. Typical block execution times are 35 ms for an AI block, 65 ms for PID, and 30 ms for AO. The configuration tool displays the macro cycle that results from the scheduling. Since the controls are executed in parallel, eight loops does not take much longer than a single loop. Only function blocks are scheduled. The execution of other blocks is controlled by other means.

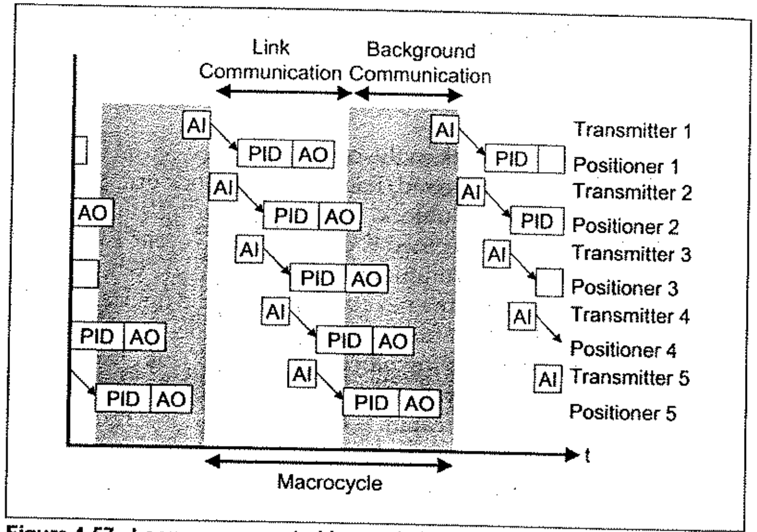


Figure 4-57. Loops are executed in parallel in different devices for optimum timing.

For the communication scheduling a window is left open for which no function block links are scheduled. This window is used for communications between the field devices and the host, which is called "background traffic." Such traffic does not interfere with the time- critical function block links used for process control. Background communication includes for example, the parameters displayed on the operation workstation screen, that is, process variables for a mimic screen or tuning parameters for a detail screen. The communication schedule on one network does not affect the communication on another network. In other words, every network in the plant has a different schedule and different macro cycle.

Every device has a built-in clock in order to follow the schedule. The FOUNDATION Fieldbus system management features assure that all device clocks are synchronized.

The macro cycle is deadtime, which is a significant component in loop dynamics and therefore must be constant. The scheduling ensures that the blocks are executed precisely, which keeps the deadtime and loop dynamics constant and ultimately makes the loop easy to tune. It is therefore essential that all devices in the loop, including any central controller (if used), are configured in the FOUNDATION Fieldbus control strategy programming language and participate in the scheduling. Unscheduled communication would result in varying deadtime and a loop that is difficult to tune. If central controllers use a proprietary language you should ensure that the execution can be synchronized with the FOUNDATION schedule.

Similarly, it is desirable to minimize the macro cycle since a shorted deadtime also means better performance and a loop that is easier to tune, especially for "fast" processes that have a short lag time, such as flow. You can minimize the macro cycle by optimizing function block execution, external links, and background traffic.

Optimizing Block Execution

Different devices take different amounts of time to execute blocks. One type of device may execute a PID block slightly faster than another. Where loop execution time is very critical it may be worthwhile to study the execution times of the blocks to find a high-performance device. Independent functions that can be executed in parallel should be distributed into separate devices so they can be executed concurrently rather than sequentially. If one device executes many blocks it will make the macro cycle longer. However, you will find that multiple variables in a loop can often be processed in parallel to some extent. Signal conditioning like flow computation, averaging, and lead-lag are examples of these (figure 4-58). It is therefore a good idea to allocate blocks that can be executed concurrently to other devices that take part in the same loop. This will enable them to be executed simultaneously.

Optimizing Function Block Links

Internal function block links do not affect loop performance. One way to reduce the number of external links is to reduce the number of devices that are involved. This can be done by only allocating blocks into the devices that are doing the measurement and

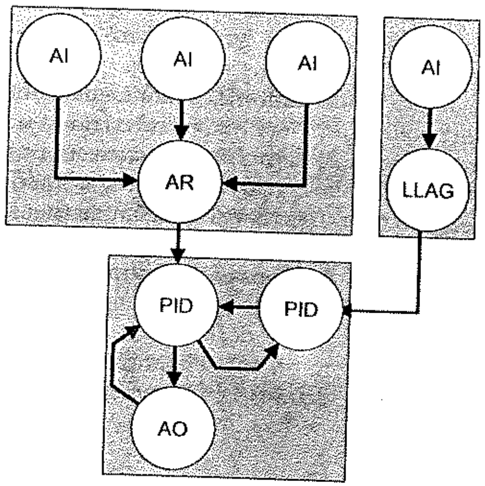
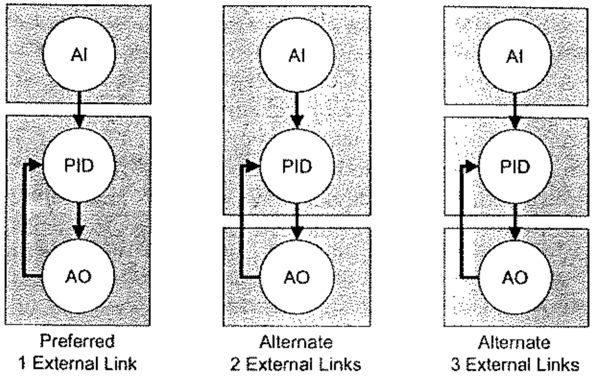


Figure 4-58. Distributing concurrent tasks in separate devices.

actuation in the loop rather than a separate device. The blocks shall be allocated in devices in such a way that links become internal rather than external. For example, the PID block in a simple PID loop can be executed in the positioner, transmitter, or in a central controller, resulting in one, two, or three external links, respectively (figure 4-59). The greater part of the macro cycle consists of function block links, which improves the performance by reducing the number of external links. In other words, for loop performance it is good to allocate the PID close to the final control element to minimize the number of external links, since these have to be communicated.

An internal link may be made from one function block input to another within the same device. A configuration tool automatically uses this capability to make internal links instead of multiple links between the same two devices. This reduces the number of links to be communicated and consequently the macro cycle. For example, if the output of one block is connected to two or more function block inputs in another device, only one external link is communicated from that output to the first input. The second and subsequent inputs are linked from the first input.

Even if a function block output is linked to function blocks in more than one other device, it is still only communicated once. This is because all devices will receive that communication at the same time as all external links are broadcast on the network for all devices to use.



Preferred
1 External Link

Alternate
2 External Links

Alternate
3 External Links

Figure 4-59. Block allocation affects the number of external links.

Optimizing Background Traffic

The larger the window that is left for background traffic, the longer the macro cycle will be. By reducing the window for background traffic you improve the control performance. Generally, the host allows the background time to be configured individually for each network. However, a smaller window for background communication means that fewer parameters can be communicated to the host each macro cycle. This therefore slows the update time for the operator screens on the host. A balance must be struck between good loop performance and good host screen updated time. If a lot of time is allocated for background traffic, the screen update will be fast, but the macro cycle will be longer. Hosts generally use several mechanisms for making the background communication efficient. View objects are groups of parameters in each function block. There are four types of view objects, which make up the static and dynamic information that is displayed in faceplates and a detail screen. The view objects ensure that more information can be received in a single communication, thus reducing communication overhead. The four views are as follows:

1. Some dynamic parameters - suitable for faceplates
2. Some static parameters - suitable for faceplates

3. All dynamic parameters - suitable for a configuration tool
4. The rest of the static parameters - suitable for a detail screen

A Multiple Variable Container (MVC) object aggregates requests for parameters from all blocks in the device and can therefore reduce the communication overhead even further. The host automatically defines the MVC based on the information that will be retrieved from the device. MVC it is a mechanism used only with devices that support this feature. It is a good idea to use MVC when retrieving information from many different blocks in a device.

As far as engineering the system is concerned it is a good idea to operate the control loop only from a single function block. This will minimize the number of view objects the host must request in order to display loop status. For example, a simple loop should be supervised to access all the data from the PID block alone, rather than some from the AI block and some from the AO block.

EXERCISES

4.1 What is the device called that passes data from one H1 network to another?

4.2 If the LRV is -200 and the URV is 500, what is the span?

4.3 If the LRV is –200, the URV is 500, the applied input is 50, and the button for setting the LRV to the applied input is pressed, what will the URV be?

4.4 In which kinds of blocks are FOUNDATION Fieldbus and PROFIBUS PA devices configured (as opposed to configuration of control strategy)?

4.5 Are the SP and OUT function block parameters static, dynamic, or nonvolatile?

4.6 How is it possible to determine remotely whether simulation is enabled in a FOUNDATION device?

4.7 In which parameter is the desired mode set for FOUNDATION and PROFIBUS PA devices, respectively?

4.8 What are the three parts of the status parameter?

4.9 Can a function block in one device be linked to a transducer block in another?

4.10 In which parameter in a standard FOUNDATION transducer block can the detailed diagnostics be found?

4.11 In which parameter in a transducer block can the main sensor measurement be seen?

4.12 To which parameter in the PROFIBUS PA AO block is the manipulated variable passed?

4.13 What do you call the capability to freely create multiple, differently tagged copies of a function block type?

4.14 To minimize the number of communicated links in a simple loop that consists only of AI, PID, and AO, should the PID block be located in the transmitter, in a shared central controller, or in the positioner?

4.15 What is the difference between an alarm and an event?

4.16 What is the setting of XD_SCALE in kPa that is required to convert the hydrostatic pressure reading in a 10 m tall tank with water (density 1000 kg/m^3) when the transmitter is mounted half a meter below the tank?

4.17 What are the two types of "Good"?

4.18 Does the fault-state option shut the loop down in the event of power loss?

4.19 Which parameter is used to configure "uncertain" so it can be used as "good"?

4.20 Which is the fail-safe mode in the FOUNDATION Fieldbus AO block and which parameter does the fault-state value set?

4.21 What mode should the block be in so as to write the input/output option (IO_OPTS) parameter?

4.22 What mode in the PID block ensures bumpless transfer when the cascade structure is broken?

4.23 Do the setpoint limits apply to the setpoint value that the operator enters in automatic mode?

4.24 If the mode must remain in manual after a problem has cleared, what configuration must then be made?

4.25 What setting is required for the AI block so pressure can be measured with a pressure transmitter?

4.26 Which output limit is more constrictive, OUT_SCALE our OUT_*_LIM?

4.27 In the PID block local override mode which parameter does the tracking value set?

4.28 Where is the PID control loop action configured?

4.29 Does the simulation function in the output blocks change the physical output?

4.30 Is the default fault-state option in the AO block availability or safety?

BIBLIOGRAPHY

1. *PROFIBUS, Draft, PROFIBUS Profile, PROFIBUS-PA, Profile for Process Control Devices*, version 3.0, April 1999, *PROFIBUS PROFILE*, Order No. 3.042.
2. ANSI/ISA-88.01-1995, *Batch Control Part 1: Models and Terminology.*
3. FOUNDATION *Specification, Function Block Application Process Part 1*, FF-890, revision 1.4, 29 June 1999.
4. FOUNDATION *Specification, Function Block Application Process Part 2*, FF-891 revision 1.4, 29 June 1999.
5. FOUNDATION *Specification, Function Block Application Process Part 3*, FF-892 revision 1.4, 29 June 29 1999.
6. FOUNDATION *Specification, Transducer Block Application Process Part 1*, FF-902 revision PS 3.0, 21 April 1998.
7. FOUNDATION *Specification, Transducer Block Application Process Part 2*, FF-903 revision PS 3.0, 21 April 1998.
8. ANSI/ISA-18.1-1979 (R1992), *Annunciator Sequences and Specifications*, Research Triangle Park, NC: ISA, ISBN 0-87664-346-2, 1992.

5

Integrate and Migrate

It will be some time before every conceivable type of device is available with FOUNDATION Fieldbus, PROFIBUS PA, and HART. Until then, systems will have to be hybrid in nature, capable of accepting both fieldbus and conventional signals.

In many plants, the existing control system has reached the end of its useful life, and must therefore be reinstrumented. It is difficult to gradually change an existing DCS into fieldbus. Changing just a single transmitter is not possible. The first step is to build a new infrastructure of networking and software. This is a large and somewhat disruptive step, but from this point on the migration can continue in smaller steps. The fieldbus technologies and the products designed around them have features to make such migration steps possible and relatively easy. When a fieldbus is added to an existing system that does not inherently support any fieldbus technology the functions not supported by the existing control system can be performed from a separate workstation, although this results in two "dashboards." Just adding a field-level network interface module to a legacy system does not make it fieldbus. The host needs supporting software and a high-bandwidth host network to benefit from new functionality like configuration and asset management. Using the process I/O alone is not enough. Lastly, the process control system must be integratable with other systems around the plant.

Hybrid I/O

For many years now, several manufacturers have made available all the main process control instruments like valves and temperature, pressure, level, and flow measurement devices in FOUNDATION Fieldbus, PROFIBUS PA, and HART versions. However, it will be some time before odd measurements unique to a particular industry become available in these versions. Conventional devices will have to be used until then.

There are two basic ways that conventional devices can be integrated with fieldbus devices within the same system. One way is to have a traditional I/O subsystem for conventional signals situated at a central location within the system. Another way is to use field-mounted converters that are connected to the fieldbus network and interfacing conventional devices (figure 5-1).

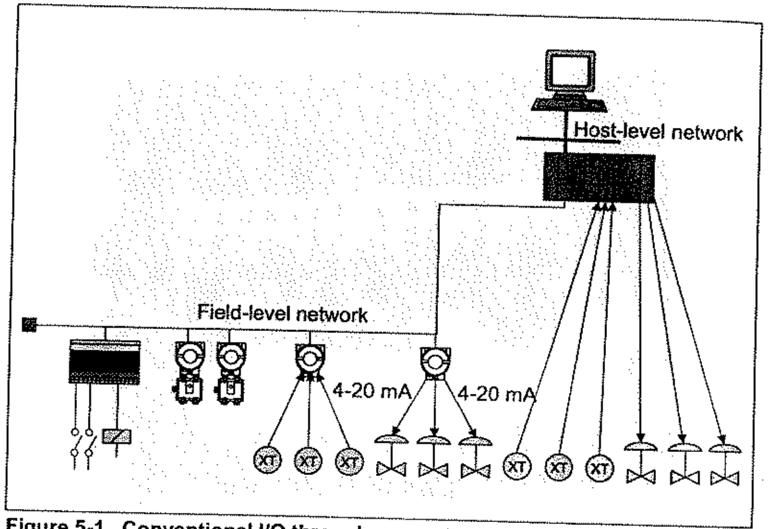


Figure 5-1. Conventional I/O through converters or subsystem.

The I/O subsystem would typically connect directly to the host-level network and could even be integrated within the linking device.

Field-mounted converters accept inputs from several 4-20 mA transmitters, for example, for vibration, power, or current, or they transmit 4-20 mA to a conventional positioner, variable- speed drive, local indicator, or the like. Converters for converting fieldbus into a pneumatic signal like 0.2-1 bar (3-15 psi) are available, that is, the fieldbus equivalent of the conventional I/P converter (figure 5-3). Low-cost pressure transmitters of suitable range can be used to accept 3-15 psi inputs.

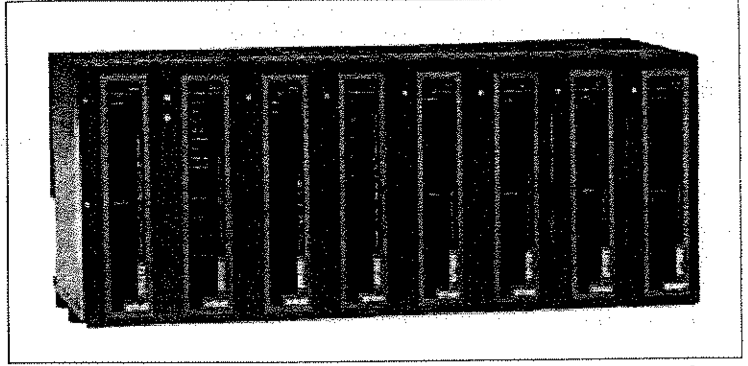


Figure 5-2. Linking device with conventional I/O modules. (Courtesy of Smar)

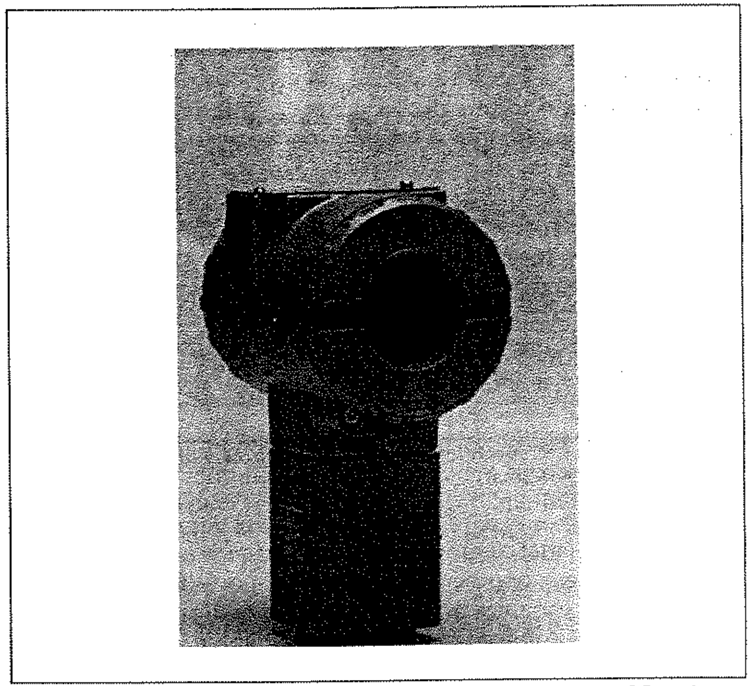


Figure 5-3. Fieldbus to pneumatic signal converter. (Courtesy of Smar)

Weatherproof converters can be distributed into the field and mounted close to the conventional devices without having to run the conventional wiring to the control room. This architecture may also make it easier to replace the conventional device and converter with a fieldbus device in the future. Signals from existing equipment can remain connected to the conventional I/O subsystem of the old system and then also connected to the new system through the converters.

More sophisticated discrete devices such as on/off valves and electrical actuators are already available that support various fieldbus technologies. Moreover, low-cost transmitters are taking the place of switches in many applications simply because devices like pressure switches do not provide sufficient diagnostics. For this reason, the need for conventional discrete I/O is reduced. Field-mounted converters for small numbers of discrete I/O are suitable for proximity switches, solenoids, and the like (figure 5-4).

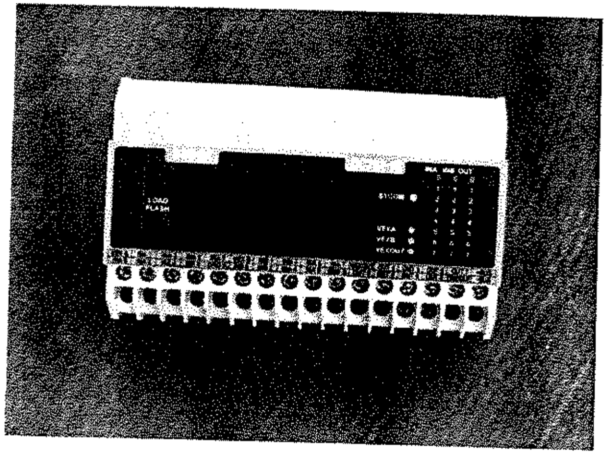


Figure 5-4. Conventional I/O mounted remotely on field-level network. (Courtesy of Smar)

Simple serial communication devices such as weighing scales, flow computers, or other remote terminal units (RTU) often provide process variables that have to be part of the control strategy. This is a typical application in which, for example, a Modbus protocol gateway can be used to map input and output registers into FOUNDATION Fieldbus or PROFIBUS block parameters.

To make the system homogeneous and the configuration of control strategy easy it is a good idea to use conventional I/O that makes the analog and discrete inputs and outputs—and even those from serial protocols—available as if they were fieldbus I/O. In this way, control loops can be implemented consistently regardless of whether they are conventional or fieldbus based. For example, if the system is using FOUNDATION Fieldbus it is easier to build control strategies if FOUNDATION function blocks also represent conventional I/O. If languages are mixed, the inherent capabilities enabled by status and the like may be lost.

Migration

Implementing new fieldbus technology is a must to enhance productivity and reduce operational cost. To maintain the old it is necessary to protect the existing investment. There is a tension here in that protecting the past investment will make it difficult to protect the investment made today and in the future. Striking the right balance between these conflicting goals will maximize the system's contribution to the bottom line. A mixed traditional and fieldbus environment is inevitable during the transition to a fieldbus technology. The benefits of introducing a fieldbus technology in a plant that already has a DCS outweigh any concerns over managing a mixed environment. Through a transition to fieldbus technology the system will be able to better cope with future changes and expansions.

A phased introduction may be necessary. As a first step, cease further investment in old and proprietary technologies. Changing any design philosophy entails risk. However, for the introduction of these fieldbus technologies the risk is manageable. The technologies have matured, and training materials and tools are available to help plants develop standards and practices before embarking on an actual project.

Taking into account future developments, consider how long an old system can continue to be upgraded or whether it is better to reinstrument to new technology now. Keep in mind that the first step of plants that have already migrated to fieldbus was simply to get familiar with the fieldbus technology. They typically did this by installing a small system in an autonomous unit of the plant, where it did not disturb the main part of the existing control system.

Reinstrumentation

Reinstrumenting an existing plant to fieldbus involves changing the field instruments, or just parts of them, to the fieldbus equivalent. Subsequently, changes also have to be made toward the host end. A complete reinstrumentation, including workstations and software, makes it possible to benefit fully from the fieldbus technologies, just as if a new system had been installed.

Many smart devices sold since the early 1990s can be upgraded to a fieldbus technology by changing the main circuit board but leaving the rest of the device unchanged. This solution is easier and cheaper than replacing an entire device. If the device is operating in a hazardous area this upgrade may have to be made by a specially qualified person. One way to minimize the costs today and prepare best for the future would be to adopt a policy in which any conventional devices purchased from now on must be fieldbus "ready" through a simple upgrade path. This will keep migration costs down and make future transition to fieldbus easier.

Fieldbus control valve positioners can replace their analog counterparts and typically any position feedback transmitter or limit switches as well since these functions are generally incorporated into a fieldbus positioner. The valve package is converted to fieldbus without having to purchase a new valve body or actuator. In many cases, the retrofitting can be done without even removing the valve from the process line. Similarly, primary sensing elements like temperature sensors and orifice assemblies can be left as is by simply replacing only the transmitter.

The fieldbus technologies were all designed to work using the same type of wires that are used by conventional instrumentation. Some of the system's existing wiring can therefore most likely be reused in a tree topology when the plant is reinstrumented. There is consequently no need to purchase or run new wiring. Even existing DC power supplies can be used to power the new instruments. All that is required are DC-powered impedance modules. As linking devices take the place of I/O subsystems and controllers some of the old panels can be utilized as well.

Upgrade

When a plant has a large investment in an existing system and expands with fieldbus instruments for the additional lines it may be hard to justify a complete reinstrumentation of the existing controls. In such cases, the best option may be to upgrade the host to handle fieldbus. Integrating a modern technology into legacy architectures is not as easy as implementing a purpose-built fieldbus system, but it can be done. Such integration is ideal for evaluating one or more fieldbus technologies on a small scale on a site that already has a control system before going all out for one of the fieldbus technologies.

The traditional solution for enabling additional communications protocols in traditional system architecture is by using a gateway.

This worked well for simple protocols used only for process I/O points and basic loop operation. However, fieldbus technologies like HART, FOUNDATION Fieldbus, and PROFIBUS PA carry such a large volume of asset management data and other information that it becomes unrealistic to map the data through a gateway. Mapping hundreds of pieces of information for each instrument is tedious and conducive to error. Moreover, the networking, database, and software in the old system most likely do not support the several magnitudes of increase in the amount of instrument information.

As part of a modernizing solution for existing systems that cannot be replaced yet, the I/O part of fieldbus can be channeled to the central controller. On the other hand, the asset management and device configuration should be handled by dedicated fieldbus hardware and software in parallel. The central controller will remain in use, and operation can continue to be performed from the existing consoles. Essentially all control systems support the Modbus protocol. Modbus can therefore be used as an intermediate protocol to gateway only the I/O parameters through the I/O subsystem (figure 5-5).

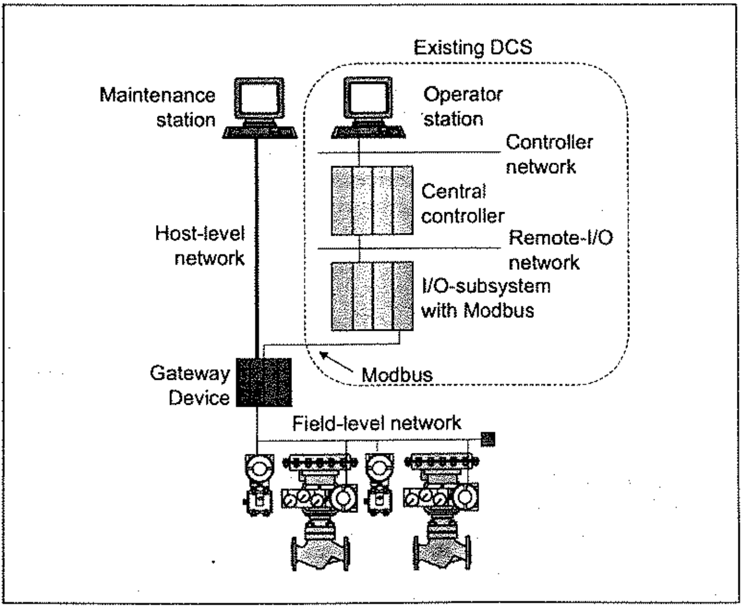


Figure 5-5. Gateway fieldbus I/O to an existing system over Modbus.

Modbus gateway exists as functionality incorporated into a linking device. A serial communication card in the existing control system polls the linking device for the fieldbus I/O parameters that were mapped into Modbus registers using Modbus commands.

Because the fieldbus functions will be carried out from a separate workstation and the central controller is configured in a different programming language, the operation environment will not be homogeneous. However, such a mixed environment is better than not having any fieldbus functionality at all.

Fieldbus in Conventional Systems

When you fit an existing or new system based on conventional technology with FOUNDATION Fieldbus or PROFIBUS PA you should take care to ensure that the integration is made tightly.

Status Propagation, Interlocks, and Permissives

Control systems designed in the past use proprietary languages that originally did not make use of the status associated with these protocols. It may therefore be a good idea to ensure that your central controller has been adapted to make use of and propagate the status. Status handling is necessary to automatically provide fail-safe action on "Bad", reset windup protection limited status or open loop and bumpless transfer when the loop is closed without custom configuration. Without this capability improved control and safety cannot be achieved.

Scheduling and Communication

A conventionally based system most likely uses a proprietary programming language that may not be synchronized with the FOUNDATION Fieldbus communication schedule. The schedule is used to minimize and dead time and to keep it constant. If strategy in the central controller is not synchronized with the fieldbus communication the sampling time and execution is not constant, dead time is not constant, and the loop is hard to tune precisely. It is a good idea to ensure that the traditional system schedules the fieldbus communication with the central controls. Ideally, the FF language should be used throughout the system.

Control Strategy Programming Language

The proprietary language in the central controller should be able to be integrated easily with the function block programming language used to configure the field instruments. Make sure that it will be easy to manage the mix of languages. To avoid confusing mapping, data should pass between blocks in the field instruments and in the central controller using the same tag throughout the system. There should be a single integrated database so changes made to the fieldbus devices do not necessitate later changes in the control strategy and vice versa.

Integration

There are two cases where a new fieldbus system must be capable of being connected to several other systems: for an expansion of a plant in which a traditional system controls the existing part of the plant and for a new installation where other systems perform functions like emergency shutdown or paper quality control. The preferred advanced controls, including safety shutdown and compressor controls, typically come from a different vendor than the supplier of the basic control system. Ideally, all subsystems should use the same host-level protocol, but since protocols in the past were proprietary that is often not the case.

Central Operation

Together, the systems around the plant form a single larger integrated system that should be operated from a central location. Generally, there is not much process I/O information exchanged between the control strategies executing in different systems. However, when the overall system is supervised from a single central location high volumes of data will have to be passed from all systems to the operator consoles. The data may be setpoints, alarm limits, tuning parameters, alarm status, process variable, and output. Since most of the data is not directly used in the control strategy there is no point routing the communication through one of the controllers. Mapping parameters for the supervision of every loop would be very time consuming. Secondly, routing data through a controller may not be good practice since there will be a common point of failure and additional load. For safety shutdown systems, such an arrangement is not allowed. Instead, subsystems are usually linked together at the host-level network (figure 5-6).

Using OPC (OLE for process control) is one way to integrate the data from the new fieldbus system that is to be presented together with the data from the other systems. Typically, a fieldbus system already comes with an OPC server, and third parties have developed OPC servers for many of the existing systems, for example, most DCS. OPC is further explained in chapter 7, "Operation."

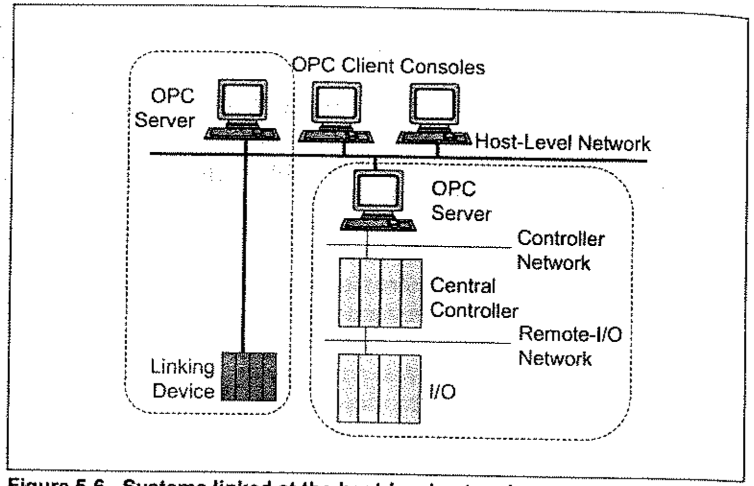


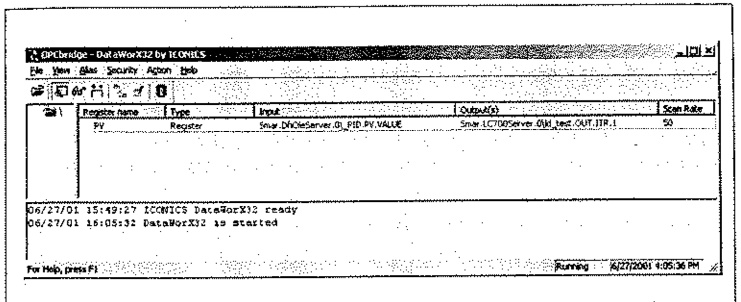
Figure 5-6. Systems linked at the host-level network.

Workstations that are OPC clients access the data from the servers of the different systems and present them together. The OPC server takes care of the data type, device address, memory register, and other hardware characteristics in order to furnish the parameters to applications in a simple format.

You can use OPC bridging software to pass some data from one system to the other, for example, from a new fieldbus system to an existing DCS. OPC bridging software reads a parameter value in one server and writes it to a parameter in another server (figure 5-7). By passing data through OPC directly to the workstations, the controller is not loaded unnecessarily. However, to do this the OPC server must be up and running at all times. If you are using redundant host-level networking, it may be a good idea to use redundant OPC servers as well.

OPC works between software applications in a workstation but also between computers on TCP/IP over Ethernet. It is therefore a good idea to use Ethernet as a platform for the host-level network since the Ethernet media can be shared by OPC, the host-level fieldbus, and other protocols. Preferably, the networking shall support both 10 Mbit/s and 100 Mbit/s because this will enable all kinds of Ethernet devices to be connected.

It is important to remember that although OPC is great for bringing data from all kinds of hardware into all kinds of software, it does not standardize the semantics of the data. When several different systems are integrated they form a larger system that includes some proprietary communications and programming languages, in which data has different meaning. You must therefore take care to interpret data correctly. For example, every system may have a different method, parameter, and code for putting a control loop into manual mode. Ideally, the different subsystems should all use the same programming language and networking. If they don't, the process visualization application has to be configured in such a way that, depending on the source of the data, it is properly converted before display.



Register name	Type	Input	Output(s)	Scan Rate
PV	Register	Smar.DfiOleServer.0_PID.PV.VALUE	Smar.LC700Server.0\ld_test.OUT.ITR.1	50

```
06/27/01 15:49:27 ICONICS DataWorX32 ready
06/27/01 16:05:32 DataWorX32 is started
```

Figure 5-7. OPC bridging of operator data.

Several of the systems that control the plant may have Ethernet connectivity, which makes it easy to physically interconnect the systems. However, remember that neither Ethernet nor TCP/IP are sufficient to enable interoperability. Many times, a proprietary protocol is used on Ethernet. OPC will therefore ultimately be needed if you want to consolidate information from the different sources in case more than one protocol is used.

Another important aspect of an Ethernet-based host-level network platform is that many legacy devices can simply use Ethernet as a medium for communicating their data without having to worry about the complexity of adding networking hardware. Rather than having several dedicated RS232, RS485, RS422, and proprietary media networks the devices can connect to a single plantwide Ethernet through a device server (figure 5-8). The device server handles hardware handshake signals as well as software functions. The data and hardware signals are tunneled to the host end where a second device server converts them back to the original signal. This makes the Ethernet communication transparent; that is, it does not necessitate any modification of the software.

Device servers from third parties are available for a number of common legacy protocols. In certain applications, a host-end device server is not required since software in the workstation replicates the Ethernet communication as a regular serial communication port, which the software accesses. You can also use device servers to connect devices around the plant that have a communication port but were never networked to Ethernet. Although this solution does not provide interoperability at the application level, it serves well to simplify the physical networking.

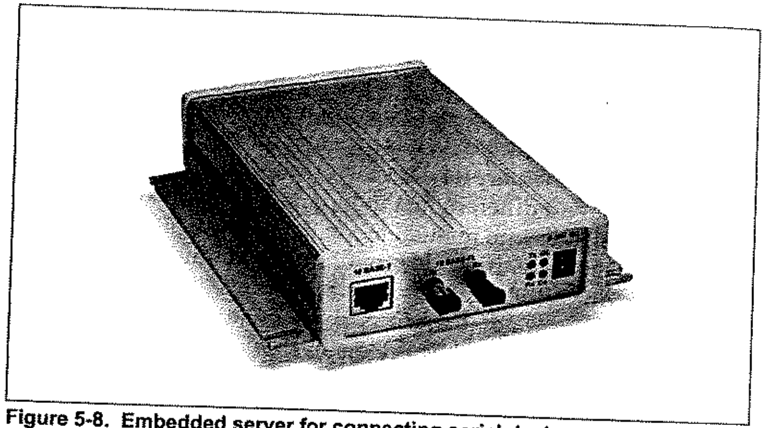


Figure 5-8. Embedded server for connecting serial devices over Ethernet. (Courtesy of Lantronix)

Subsystem Considerations

The controls performed by a DCS, PLC, or a FCS are usually referred to as "basic process control." A plant may have more than one basic process control system. For example, perhaps the existing system could not be integrated with fieldbus technology, so a new expansion had to be made using a modern system. In addition, some plants require shutdown systems and advanced controls as well as other functions provided by auxiliary systems. In this case, various subsystems around the plant can be connected to a plantwide Ethernet, which serves as the platform for the host-level protocol and OPC. Dated systems for basic controls as well as new advanced process control can be fitted with OPC servers to make the data exchange possible.

Modbus is a simple protocol supported by many existing systems that can be interfaced to a fieldbus system, for example, through an HSE gateway. If it is not possible to network subsystems a last resort is to "hardwire" them together using conventional I/O. This is not an ideal solution, but it works if only a few signals are involved.

Existing Basic Controls

Because fieldbus is different from traditional controls, introducing it into an existing plant may be viewed as a challenge. However, managing a mixed-system environment is significantly less complex than the challenges plants have faced in the past. Before, plants likely had a mix of incompatible protocols that had to be integrated using expensive interfaces and third-party drivers. It was like a control room where operators speak different languages. Managing this environment was an exercise in complexity, which in turn drove up costs.

Though fieldbus and DCS are technically different the operator at the workstation can work with them in the same basic way. An operator who is familiar with DCS will find that he or she already knows how to change modes and setpoints or view process variable and alarms in a fieldbus-based system. Because of the commonalties between the basic operation of DCS and fieldbus, managing a mixed environment is a less formidable task than it might appear. Supervisory data is drawn from the existing DCS fitted with an OPC server and from the new fieldbus system. It is then displayed at the same workstations using a process visualization software as an OPC client.

Advanced Process Control

For a vast majority of controls and monitoring plants can build the control strategy using function blocks, ladder diagrams, and other regulatory and logic programming languages as found in standards like FOUNDATION Fieldbus and IEC 61131-3. They can also use many proprietary programming languages in DCS and PLCs. Usually, such a control loop never utilizes more than three or so process variables. However, some applications, primarily in the chemical and petrochemical industries, are highly interactive and run against multiple constraints. For such control loops, advanced process control (APC) can further optimize the controls. The APC system is essentially sophisticated process analysis and control software that also has a user interface application. The APC algorithm utilizes perhaps a hundred process variables and provides scores of manipulated variables. One of the main reasons to reinstrument a plant is to enable multivariable advanced control.

As in the past, the APC connects with the basic process control system to form an integrated environment. The APC does not need dedicated field instruments; rather, the APC is able to utilize the information from the transmitters in the basic control system. Similarly, the APC actuates by using valves and other final control elements connected to the basic control system. In other words, in a system with fieldbus the APC system uses fieldbus networking and devices to perform the measurements and provide the actuation. Traditionally, the APC had a custom interface to the particular type of proprietary DCS used in the plant. Alternatively, the APC had an interface to the plant historian software, which in turn had a custom interface to the DCS. These custom interfaces were expensive, and a historian interface usually necessitated parameter mapping, which resulted in duplicated databases. Using direct OPC interfaces between the APC and fieldbus control systems is both inexpensive and eliminates mapping, resulting in a single integrated database. Many APC systems now support OPC.

Shutdown System

In certain industries, safety shutdown must be performed on some of the loops. It is performed by an emergency shutdown (ESD) system, which is connected in parallel with the basic control system for certain loops and monitors the process for hazardous conditions. If such conditions are detected it shuts the loop down. The ESD is typically a single, dual-redundant or triple-redundant PLC with a wider diagnostic coverage for which a safety certification has been obtained. As with DCS in the past, safety systems must have their own dedicated field instrument and logic. This is so because hardware and software cannot be shared since no common mode of failure can exist between the basic control system and the ESD. Many ESD already support OPC on TCP/IP over Ethernet.

All ESD use 4-20 mA to interface with process instrumentation. A number of proprietary bus technologies have been approved for use in ESD as well as the PROFIsafe profile of PROFIBUS. However, so far no process instrument or ESD for PROFIsafe is available, even though it offers clear advantages over 4-20 mA. For example, unlike 4-20 mA, PROFIsafe uses the superior diagnostics capabilities possible in standard digital communications.

For more details about shutdown systems refer to the book *Control Systems Safety Evaluation and Reliability* by William M. Goble (2d ed., ISA, 1998).

Other Systems

There is more to running a plant than process control. Other important functions that need to tie in with the control system include production planning, materials management, quality management, plant maintenance, financial accounting, and human resources, among others. These are handled by enterprise resource planning (ERP) systems. As in the past, the ERP is separate from the process controls, but it now often ties in with the fieldbus system. Other typical systems include historians and plant information.

EXERCISES

5.1 Can conventional DC power supplies be used for IEC 61158-2 networks?

5.2 Does OPC specify the semantics of the data?

5.3 Is using OPC with shutdown systems permitted?

BIBLIOGRAPHY

1. Goble, William, *Control Systems Safety Evaluation and Reliability*, 2d ed. Research Triangle Park, NC: ISA, 1998, ISBN 1-55617-636-8.

6

Troubleshooting

When devices and controls cannot get started fieldbus technologies give you the power to use software to locate and troubleshoot problems all the way down to the sensors and actuators from a central location. Once the installed networks and devices are operational and the control strategy has been debugged the system will give you little trouble. Day-to-day calibration and maintenance for dealing with wear and tear are covered in chapter 9, "Maintenance and Asset Management."

One possible potential fault is that the communication is not working. Although the software may indicate this it may not be of much further help. Even when using fieldbus technologies there are problems for which you will have to troubleshoot with multimeter and screwdriver.

Device

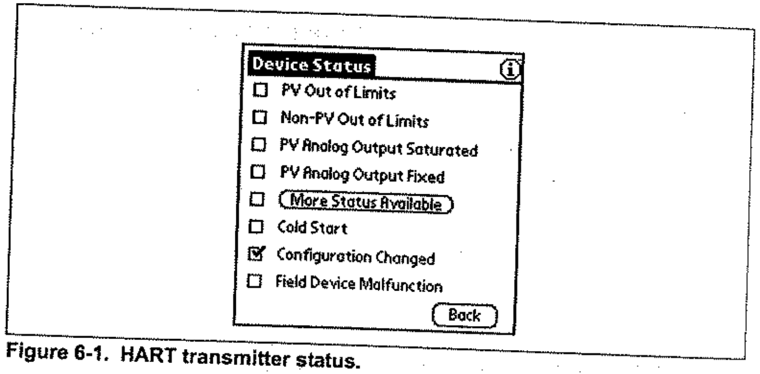
If the network is OK then communication is possible and software can be a great help in detecting faults and troubleshooting. To fully benefit from the fieldbus technologies plants should permanently connect the engineering tool to the field devices. In other words, plants should not only use the communication for process I/O like measurement and desired valve position, but also to continuously monitor device health. It is also a good idea to have the engineering tool running at all times so any troubled device in the plant can be checked quickly.

HART Devices

General Field Device Status

Whenever the field device communicates with the handheld or host it always reports back its status. In installations that have permanently connected communications interfaces checks are continuous, and any problems are reported immediately. However, HART installations are predominantly using only the 4-20 mA signal, and a handheld is only connected occasionally for short time. In these cases, the associated controller must monitor the 4-20 mA to detect faults and prompt the user to connect a handheld terminal. A current below 4 mA or exceeding 20 mA usually is a general indication of a fault.

Once such a general fault is detected you can connect a handheld terminal to carry out detailed interrogation. There are a number of standard diagnostic messages (figure 6-1). The device's manual would typically give more specific clues as to the nature of the problem associated with each diagnostic message.



Device Status

☐ PV Out of Limits
☐ Non-PV Out of Limits
☐ PV Analog Output Saturated
☐ PV Analog Output Fixed
☐ More Status Available
☐ Cold Start
☑ Configuration Changed
☐ Field Device Malfunction

Figure 6-1. HART transmitter status.

Field Device Malfunction

The Field Device Malfunction message indicates that the device is unable to perform its function. For a transmitter, it is generally an indication that the primary sensor, auxiliary sensor, or the sensing circuitry has failed. For a valve positioner or other final control element it is usually an indication that the valve is unable to move.

Configuration Changed

Configuration Changed is simply an indication that the device configuration has been changed. Most handheld terminals don't display it. It is primarily intended for use by a host (primary master) to detect that a handheld (secondary master) has changed the device configuration. This status prompts the host to update its database to the device configuration to ensure that the device configuration file is synchronized with the actual device.

Cold Start

Cold Start is an indication that power was removed from the device and that it is once again powered up. It means that the device may have been removed and reconfigured separately. It may also mean that the device does not have enough power supply to operate properly and is therefore switching off and on.

More Status Available

The general device status provides an overview of the health of the instrument. Detailed diagnostics must be retrieved separately. The More Status Available flag indicates that more detailed diagnostics is available and should be checked.

Analog Output Current Fixed

Analog Output Current Fixed means that the 4-20 mA output is in a constant current mode, for example for loop testing or simulation. That is, it indicates that the output does not correspond to the input applied to the sensor. It is also displayed when the 4-20 mA is deactivated in multidrop mode.

Analog Output Saturated

Most 4-20 mA transmitters follow the NAMUR NE 43 standard, in which the output follows the input from as low as 3.8 mA up to 20.5 mA, that is, from -1.25 percent to 103.1 percent of the set range. Should the input be higher or lower than these values the output current will saturate—it is limited in 3.8 and 20.5 mA and no longer follows the input. The Analog Output Saturated status means that the output is saturated in either extreme. This is not really an error; it just means that the range of the transmitter has been set too small for the application.

This status frequently appears in pressure transmitters on a test bench when zero pressure is applied to a transmitter with lots of elevation or suppression, that is, when zero is not within the set range. A classic case is the saturated output that is seen on a low range absolute pressure transmitter exposed to atmospheric pressure. For greater safety, it is a good idea to use transmitters and controllers that all support the NAMUR NE 43 standard to ensure that status can be propagated through the control strategy.

Nonprimary Variable Out of Limits

Smart transmitters usually have secondary and even tertiary sensors whose values are used to compensate for and correct the value from the primary sensor. For example, a pressure transmitter typically also measures the temperature of the sensor module in order to perform temperature correction. The Nonprimary Variable Out of Limits status indicates that one of these auxiliary sensors has reached its limits—the value is outside the compensation range of the device and the primary value may therefore not be completely accurate. It may also mean that the device is exposed to ambient stresses beyond its limits, which could damage the instrument.

Primary Variable Out of Limits

Every sensor has a limited operating range. The Primary Variable Out of Limits flag indicates that the applied input is beyond the range of this sensor. Thus, this status means that the sensor selected is not suitable for the application and that a greater sensor range must be selected.

Detailed Field Device Status

In addition to the generic field device status that is reported whenever the device communicates, many handheld terminals and devices support detailed device diagnostics. These checks are proprietary, and you should verify the problems they report against the device manual.

Simulation

The 4-20 mA loop current represents 0 to 100 percent of the range set in the transmitter. Any device that receives the signal has to be scaled accordingly to give an engineering unit readout. For example, if the transmitter range is set to -200 to +850°C the same range must be set in the receiving devices. Inconsistent ranging will result in operational problems and potentially dangerous situations. HART transmitters have a loop test function that is used to remotely set the output current independently of the applied input (figure 6-2). The loop test makes it very easy to verify readings in indicators, recorders, and controllers, thus simplifying the troubleshooting of scaling.

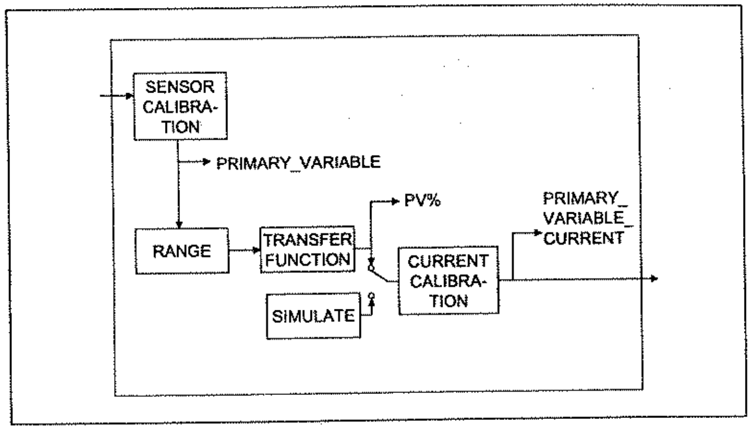
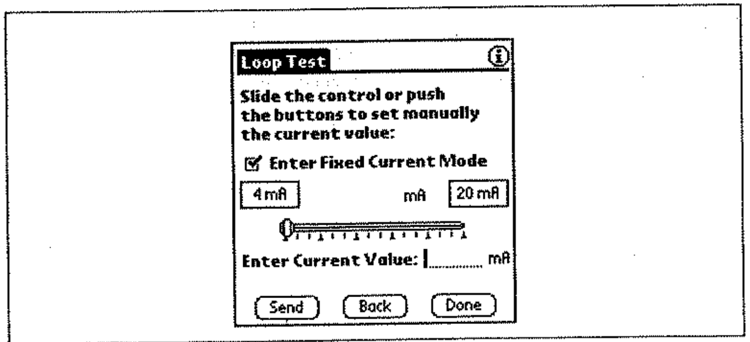


Figure 6-2. HART transmitter block diagram.

Typically, the loop current can be set within the range of the NAMUR NE-43 standard, that is, 3.6 to 21 mA (figure 6-3).


Loop Test

Slide the control or push the buttons to set manually the current value:

☑ Enter Fixed Current Mode

Enter Current Value: ______ mA

Figure 6-3. Using loop test to simulate process variables from a handheld.

You can also use the loop test to safely simulate process conditions in order to test alarm trips, controller action, and automatic shutdowns, among other actions.

PROFIBUS PA Devices

A large number of the parameters in the physical, transducer, and function blocks are there for the specific purpose of troubleshooting. These parameters are powerful troubleshooting tools.

Physical Block

Standard diagnostics for the device as a whole is shown in the diagnostics (DIAGNOSIS) parameter, and additional self-diagnostic checks continue in the manufacturer-specific diagnostics extension (DIAGNOSIS_EXTENSION) parameter in the physical block. The short diagnostic messages give a pretty good general idea about the fault. The device's manual would typically give specific clues about the nature of the problem that the various diagnostics messages indicate for the specific device.

Configuration Not Valid

Invalid configuration status indicates that one or more parameters have been configured to an illegal value. Common problems include a mismatch in the configuration of transducer blocks and their associated function blocks, particularly engineering units. Another common mistake is configuring values outside of the allowed limits. Keep in mind that ranges may vary by sensor type and that set alarm limits may become invalid when the range is changed.

Analog Output Block

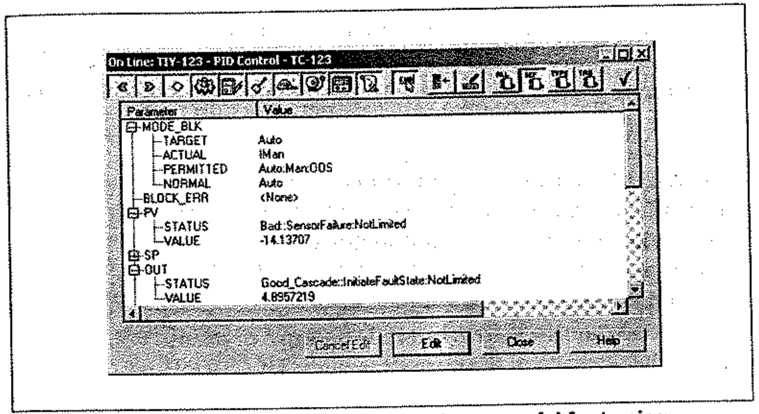
A summary of the output error status in the device is kept in the check back (CHECK_BACK) parameter in the analog output block. The short diagnostic messages give a pretty good general idea about the fault. The device manual usually associates each diagnostic message to the nature of the problem for the specific device.

FOUNDATION Fieldbus Devices

A large number of the parameters in the resource, transducer, and function blocks are present for the specific purpose of troubleshooting and constitute powerful tools. All blocks have a block mode (MODE_BLK) parameter and some parameters with status. If the actual mode is not the same as the target mode this indicates that something is wrong. Therefore, the mode parameter is a good starting point when troubleshooting (figure 6-4). The block error (BLOCK_ERR) parameter is also found in every block. You should check it when troubleshooting because it lists many possible problems.

Resource Block

The resource state (RS_STATE) parameter reflects the status of the control strategy. If the resource block mode is Auto, the Resource State is Online. In OOS mode, the Resource State is Standby. If the Resource State is Failure it means that a memory or other hardware failure has been detected.



```
MODE_BLK
  TARGET       Auto
  ACTUAL       !Man
  PERMITTED    Auto:Man:OOS
  NORMAL       Auto
BLOCK_ERR      <None>
PV
  STATUS       Bad:SensorFailure:NotLimited
  VALUE        -14.13707
SP
OUT
  STATUS       Good_Cascade::InitiateFaultState:NotLimited
  VALUE        4.8957219
```

Figure 6-4. Abnormal status and actual mode are useful for tracing problems.

Transducer Block

The transducer error parameter is found in all transducer blocks. It reports errors associated with the device as a whole. The errors reported by the transducer error (XD_ERROR) parameter are unique to each manufacturer's device. This parameter displays only a single error at a time. If several errors are present in a device simultaneously, only the highest-priority (most severe) error is displayed. To find the interpretation and remedial action for such an error you have to look in the relevant device's manual.

Communication

Communications problems caused by installation may be noticed when the system is being commissioned. Once these problems have been rectified they usually do not occur again unless changes are made to the network. For example, rerouting the cable or adding or replacing a device may introduce a grounding point. Expansion or modification may reintroduce installation problems. Other installation problems include water filling caused by improperly sealed electrical conduit connections at field devices and junction boxes.

Basic troubleshooting rules apply. For example, if there are several problems look for patterns to try to determine a common cause. Is the problem affecting one particular model of devices, or just devices on a particular network? Most importantly, what was done before the communications problems started? If a particular device appears to be causing the communications trouble, connect just that one device on a separate network and try to establish communication.

HART Communication

The system's devices have to have sufficient voltage to operate properly even when the current is at its maximum, that is, 20 mA. It is quite possible that the communication works OK when the loop current is low, but fails because of a large voltage drop along the wire when the current is high. The HART standard specifies no minimum operating voltage so you must verify the voltage against the manual for the relevant device type. If the device has a digital display, it may tell you if the device has power, but this may not be conclusive. Make sure that the device has sufficient voltage to operate properly even when the loop current is at its maximum.

The most common HART communication problem is that the handheld battery is too low and needs changing or charging. Another possible problem is that the handheld is not connected at the right place. For example, the handheld may have been connected on the power-supply side of the resistor rather than on the device side.

Multidrop Addressing

For a HART device in a multidrop mode you should set the polling address manually from the handheld. There is a chance that two devices have accidentally been given the same address, causing a communications problem. Verify the address of the device by connecting only a single device to the handheld. Usually, the handheld has a function that checks all polling addresses from 0 to 15 to verify if a device is present for any address.

Masters

A not-so-common problem is master conflict. A network can have one primary and one secondary master. The central controller or a RTU is usually the primary master, while the handheld is usually a secondary master. Some interfaces can be configured as either primary or secondary master. Two of the same could therefore occur. Verify that only a single primary and a single secondary master is present on the network. If necessary reconfigure any duplicate masters.

PROFIBUS PA and Foundation Fieldbus

When a communications fault is detected use the configuration tool to determine if it affects the entire network or just a single device (figure 6-5). If you are using repeaters or repeating safety barriers there is also the possibility that one of the network segments is affected while the others are OK.

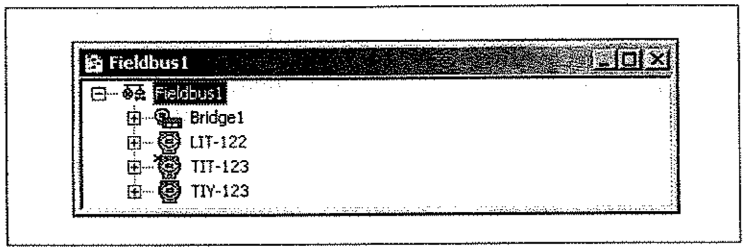


Figure 6-5. Single device communication fault.

The engineering tool usually has a "live list" that shows all the communicating devices. Ideally, you should confirm that all the devices have been connected at commissioning. Should any device fail to communicate it drops out of the list. For installation problems, the communication is often not working, and software may therefore not be much help. There are a number of checks that can be carried out using traditional tools such as a multimeter. Check continuity, sufficient voltage, and noise. Measure the resistances to detect short circuit: conductor-conductor, conductor-shield, conductor-ground, and shield-ground. It may even be a good idea to verify the cable capacitance.

Device Power

If the supply voltage at the device terminals is lower than 9 VDC the device may not operate properly. Some devices may need even more than 9 VDC. Low voltage is primarily a problem in intrinsically safe installations or even in regular installations in which long, thin wires are used. Another potential source of trouble is devices that have high current consumption. Lastly, voltage drop may also be caused by poor connections in intermediate junction boxes.

Make sure that the voltage at the device terminals is a minimum of 9 VDC, preferably more. Ensure that the power supply provides sufficient power for the combined current consumption of all the devices on the network. Check for voltage drop at the entrance and exit of junctions.

Magnetic Coupling Interference

In surroundings that have strong magnetic fields, the fieldbus network and the earth together may form a coil in which AC current is inducted by the magnetic field. This coil is formed because field instruments generally have filters that cause some capacitive coupling between their network terminals and their housing. The housing in turn is connected to earth by an earth wire or due to mechanical installation forming a loop (figure 6-6). The larger the loop, the greater the chance that noise can be picked up from the magnetic field. In a well-functioning network the voltage between the earth of the field instrument and the shield is usually no more than 0.2 VAC RMS. However, communication errors may result if the voltage is closer to 3 VAC.

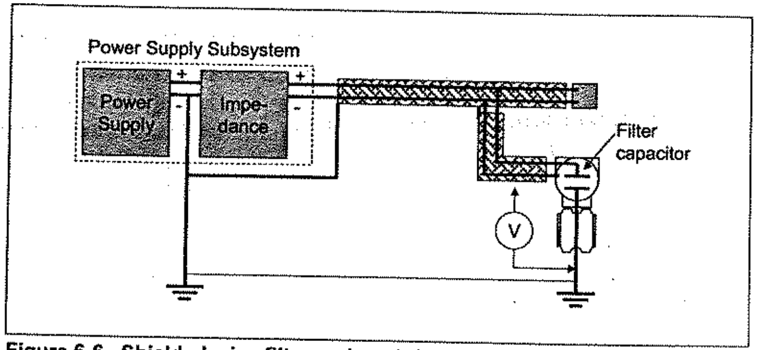


Figure 6-6. Shield, device filter and earth forming a large area magnetic pickup coil.

Check the voltage between the shield and the earth by using an AC voltmeter, preferably true RMS. Reducing the area that is enclosed by the coil will reduce the problem of noise pickup and subsequent communication loss. You can reduce the area by maintaining the cabling inside earthed metal conduits, which maintain electrical continuity for all sections of the conduit. Alternatively, you can connect an additional return conductor, preferably a bare one, within the cable to create a minimal area loop. Such a bare wire is often found inside cables (figure 6-7).

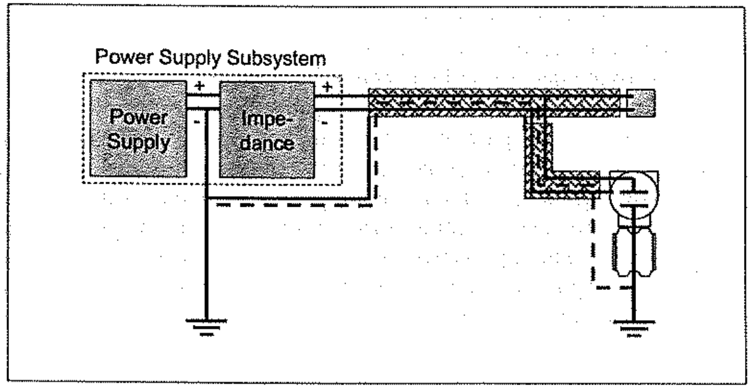


Figure 6-7. Minimizing magnet pickup.

Ground Loop Interference

The capacitors on the network connection of the earthed field instruments also act as a coupling to the earth at the device. If there is a ground potential difference between the host end and the field device end a ground loop will be formed that may disturb the communication. A ground loop may also be caused when shield or wires are accidentally connected to earth at multiple points somewhere along the cable route.

When field devices are properly connected to earth ground potential differences are reduced, which reduces the chances of communications problems. Open the earth connection at the host end (which should be the one and only earth connection) and then measure the impedance between the conductors and shield to earth. This will enable you to detect the presence of multiple grounding points. If no additional points are earthed the impedance will be measurable in terms of mega ohms. It is not uncommon for shield from poorly terminated cables to be in contact with the conduits or instrument housings.

Network Topology

On a fieldbus network, the communication is achieved by a rapidly changing signal that is essentially based on the frequency 31.25 kHz. As a result, the network impedance shall be resistive insofar as possible since any reactive characteristics cause signal degradation. For this reason, it is preferable to use a trunk topology since any spur or other deviation changes the characteristic impedance.

Check that the network topology follows the basic rules for spurs, length, device, and so on.

Earth

Heavy-load equipment must not share earth wiring with instrumentation since this may cause communication problems when the loads are switched on and off. Separate earth wires should be routed for instruments, although eventually you should earth (ground) them at the same point as all loads in the plant.

Addressing

FOUNDATION Fieldbus hosts detect new devices on the network by constantly polling the different network addresses to see which addresses correspond to an actual device. The devices that have been detected are displayed in the live list of the configuration tool or by other means. Because a network has no more than thirty-two devices, the host skips several addresses to detect the devices more efficiently. For most plants, this will never pose a problem because a system will not assign an address it does not poll. However, if a plant has two different types of systems and a device is moved from one to the other there is a slight chance that one of the systems is configured to not poll the particular address that the device coming from the other system occupies. If a device connected to a network does not appear in the live list, it could be that it has an address that is not being polled.

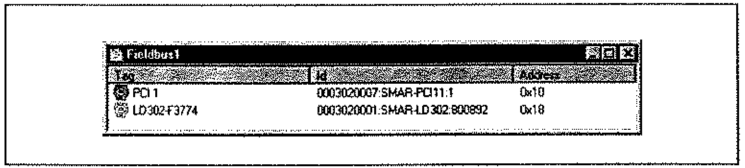
In the event the poll sequence setting has to be changed this can be done in the system management part of the configuration tool at the DlmeLinkSettings parameter. This parameter is where the element FirstUnpolledNodeId determines the first address to be skipped and NumConsecUnpolledNodeId determines the number of addresses being skipped. Addresses 0 through 15 are never polled. For example, if FirstUnpolledNodeId = 48 and NumConsecUnpolledNodeId = 184 then the addresses 16-47 and 232-255 are polled.

For PROFIBUS PA devices the polling address is set manually on the device or set from the configuration tool connected on a network that has a single device. It is also possible for it to be set using DIP switches. There is a chance that the communication problem is the result of two devices having been accidentally given the same address.

Verify the address setting of the devices. It is a good idea to keep a cross-reference list of the addresses for each of the devices on each network.

Interoperability

The configuration tool generally has a "live list" or equivalent that shows the communication status of the devices. When a device is connected the linking device or interface will first detect it and then establish communication with it. After detection the device is typically shown in the configuration tool's live list differently from fully operational devices until the communication is fully established (figure 6-8). If there are any interoperability problems between the host and the field device the communication may not be fully established, as the live list indicates.



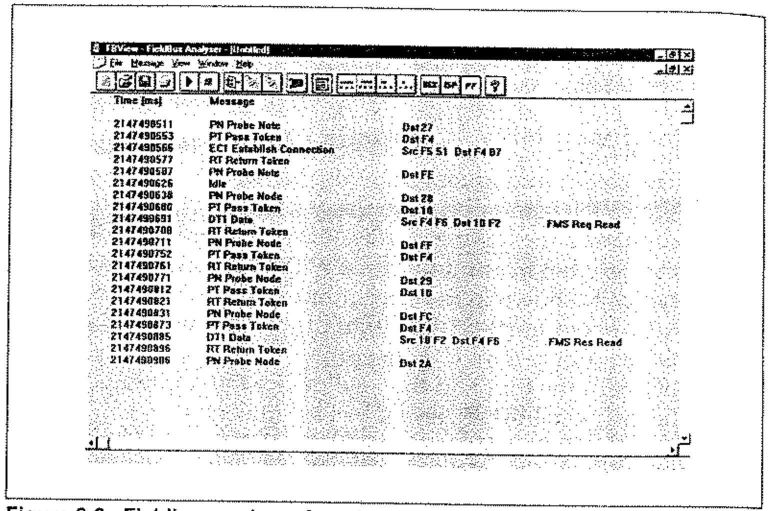
Tag	Id	Address
PCI 1	0003020007:SMAR-PCI11:1	0x10
LD302-F3774	0003020001:SMAR-LD302:800892	0x18

Figure 6-8. Establishing communication.

If there are any ways to reset the device to its original factory configuration it may be a good idea to do so because personnel may have set incompatible communication parameters in the device for some other application. If that does not work look for specific tips in the users' manual for the host.

Advanced Communication Troubleshooting

A fieldbus network analyzer passively monitors the network communication and allows it to be analyzed in detail (figure 6-9). It is not a tool that most people would use. In fact, it is rarely required. The bus analyzer is a tool for advanced users who have an in-depth knowledge of the fieldbus technology that is being used. However, most bus analyzers allow a nonexpert to easily sample snippets of communications, which are saved to a file. The file can then be sent to a communications expert for analysis. If this advanced communication troubleshooting technique is used, it is only during the commissioning phase of a plant when communication problems are encountered. Once the communication problems have been fixed, they don't reoccur unless key devices are changed.



```
Time [ms]     Message

2147490511    PN Probe Note                 Dst 27
2147490553    PT Pass Token                 Dst F4
2147490566    ECT Establish Connection      Src F5 51  Dst F4 B7
2147490577    RT Return Token
2147490587    PN Probe Note                 Dst FE
2147490626    Idle
2147490638    PN Probe Node                 Dst 28
2147490680    PT Pass Token                 Dst 10
2147490691    DT1 Data                      Src F4 F5  Dst 10 F2      FMS Req Read
2147490700    RT Return Token
2147490711    PN Probe Node                 Dst FF
2147490752    PT Pass Token                 Dst F4
2147490761    RT Return Token
2147490771    PN Probe Node                 Dst 29
2147490812    PT Pass Token                 Dst 10
2147490821    RT Return Token
2147490831    PN Probe Node                 Dst FC
2147490873    PT Pass Token                 Dst F4
2147490885    DT1 Data                      Src 10 F2  Dst F4 F5      FMS Res Read
2147490896    RT Return Token
2147490906    PN Probe Node                 Dst 2A
```

Figure 6-9. Fieldbus analyzer for advanced troubleshooting.

Ethernet and IP

When you cannot establish communications over Ethernet a good first check to perform is to use the Ping command to see if the basic communication with the device is working. If the result appears instantly and it is "Destination host unreachable," there is most likely a mismatch in the address and subnet configuration in the computer from which the pinging is being done. Another reason is that the driver for the network card driver is not installed. If the result is "Request timeout" there is either a network problem or the address setting in the Ethernet device is wrong.

Computer network interface cards, Ethernet field devices, and network hardware such as hubs, switches, and bridges usually have two LEDs. One LED, which is usually called "Link," indicates that the device has been connected to the Ethernet. The other LED, usually called "TX," indicates when the device is transmitting. These LEDs can be used to verify that the device is properly connected and whether it is responding. Network testers find electrical problems such as broken or shorted wires and "split pairs," among other issues. Cross talk and return loss are also measured. For 100 Mbit/s Fast Ethernet, the cable must be in good condition.

Ethernet switches and routers usually have built-in SNMP (Simple Network Management Protocol) agents that collect statistics about the communication. These statistics can be used for troubleshooting, for example, to see where communication bottlenecks are, to determine which devices are not performing, for utilization data, and the like. These agents can be assigned an address and interrogated remotely by network management software to present the statistics. SNMP OPC servers are also available that make it possible to display the network's overall status in, for example, the process visualization software.

Advanced users can use Ethernet protocol analyzers to see the messages that are being exchanged. However, in a switched network architecture often only the information that pertains to one device travels on any one cable, and corrupted messages are filtered out by the switch. Complex problems will not be revealed by connecting an analyzer. Often these complex problems can be overcome by configuring the switch to copy messages to the port where the analyzer is connected.

Control Strategy

If your site uses the FOUNDATION Fieldbus programming language to build the control strategy you can tell a lot about what is going on in the control strategy by checking both the block mode parameter and the status that some parameters have. They therefore make great troubleshooting tools.

One common problem is that changes to a parameter are not being accepted. If it is not possible to write a parameter it could be because the block is in a mode that prevents the parameter from being written. Some parameters require the block to be in Manual or OOS mode. It could also be that the value is outside the scale that it is associated with. Check the actual mode of the block and the setting for the associated scale parameter.

Another possible problem may not be a problem at all. It may be that the mode has shed because the fail-safe function or the reset windup protection have been activated.

If the output is not changing it may be because of the mode of the block, because some limit has been reached, or because a parameter like GAIN has been configured to zero.

Using Parameter Status

By looking at the status of inputs, outputs, and contained parameters you can often determine whether or not a problem comes from any particular block. The input status is one of the factors that control the mode of a block; that is, the input status may make the mode shed. For example, a "Bad" quality usually sets Man or IMan mode. A "Not Invited" substatus also sets "IMan" mode.

If block input is "Bad" or "Uncertain," trace the signal back to its source and check what is going on in the originating block. The substatus usually provides a clue about what the problem is. Most substatuses need no further explanation. For the back-calculation input (BKCAL_IN), it is possible to see if the lower block is forcing the block you are looking at to IMan mode. The configuration tool will typically show which block and parameter is the source for the input. This makes it easier to trace the signal backward and also indicate value and status directly in the control strategy (figure 6-10).

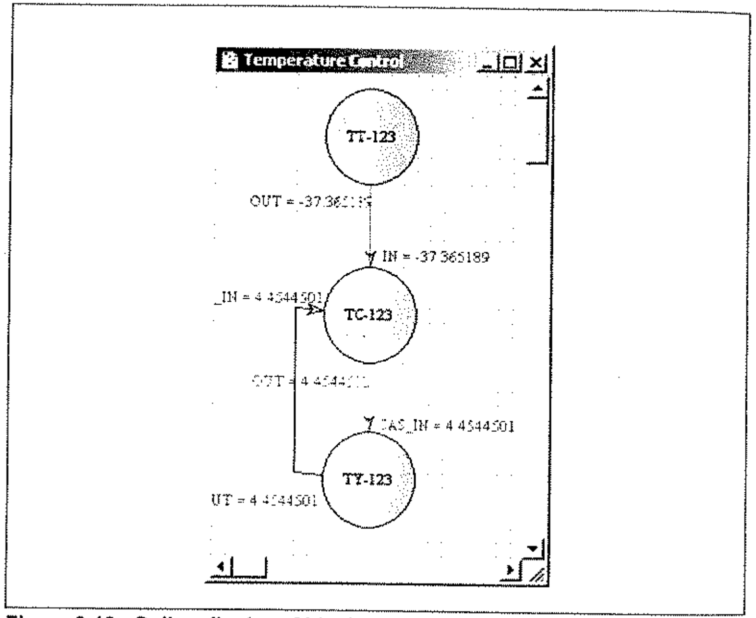


Figure 6-10. Online display of block I/O with color-coded status indication.

Using the Block Mode Parameter

If the Actual block mode does not match the set Target mode there are a number of possible reasons, depending on the Actual mode:

Out of Service

- The resource block may be in out-of-service mode.

- Some parameter has an invalid selection or a value out of range. Check the block error (BLOCK_ERR) parameter for a clue.

It is common to forget to configure the CHANNEL, SHED_OPT, and BYPASS parameters, which results in a block error and OOS mode. To eliminate these simple pitfalls, it is a good idea to use strategy templates insofar as if these are provided with the configuration tool.

Initialization Manual

- Back-calculation input status is "Bad," for example, because of a missing link or no communication.
- Back-calculation input status is "Fault State Active."
- Back-calculation input status is "Not Invited," for example, because the lower block is in the wrong mode or fault state or because there is no communication between the output and cascade setpoint input.

Local Override (Control Class Blocks)

- The output tracking input (TRK_IN_D) is True, and tracking has been enabled in the control options. If the Target mode is Manual, then this must also have been enabled in the control options.

Local Override (Output Class Blocks)

- The fault state is active based on the cascade setpoint input status or hardware. Verify in the block error (BLOCK_ERR) parameter.
- The update for CAS_IN has timed out because there is no communication, exceeding FSTATE_TIME.
- The fault state is forced in the resource block. Check the FAULT_STATE parameter.

Manual

- The block is on its way out of OOS mode. Just wait a few seconds and see if this is the case.
- The primary input (IN) status is "Bad," and bypass is not active. Check the source block.

- The primary input (IN) status is "Uncertain," the status option to treat "Uncertain" as "Good" is not enabled, and bypass is not active. Check the source block.
- The update for RCAS_IN or RCAS_OUT has timed out because there is no communication, exceeding SHED_RCAS and SHED_ROUT, respectively.
- Check the two function block links and the status of the inputs in the cascade structure.

Automatic

- The cascade setpoint input (CAS_IN) status is "Bad."
- The update for RCAS_IN or RCAS_OUT has timed out because there is no communication, exceeding SHED_RCAS and SHED_ROUT, respectively.
- Check the two function block links and the status of the inputs in the cascade structure.

Cascade

- The update for RCAS_IN or RCAS_OUT has timed out because there is no communication, exceeding SHED_RCAS and SHED_ROUT, respectively.

Using the Block Error Parameter

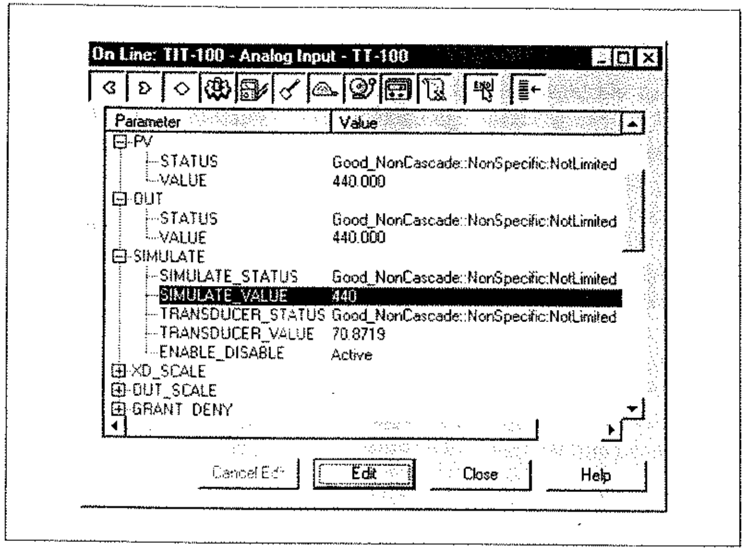
The FOUNDATION resource block, transducer blocks, and function blocks all have a block error (BLOCK_ERR) parameter. This parameter gives a general idea why a failed block is not operating the way it should. Most errors need no further explanation. One of the many errors tracked by this parameter is an error in the configuration of the device or strategy.

Block Configuration Error

A block configuration error usually means that some parameter has an invalid value. Check that the minimum set of parameters in the block has been configured, that is, all the parameters with an invalid default value. Make sure that all limit parameters are within the range set by the corresponding scaling parameter. Changing the engineering unit may make some values go out of range if they are not updated to match the new unit. This will cause configuration error.

Simulation

When troubleshooting the control strategy the simulation capability in input and output class blocks can come in very handy. It allows you to test how the control strategy behaves during process and device conditions that cannot be tested safely in reality. For example, by simply simulating a value it is possible to test the response to a high temperature or pressure or other variable without actually changing the process (figure 6-11). It is also possible to simulate any status such as "Bad" so as to check the reaction to a sensor failure. Similarly, if a sensor or actuator has failed it is possible to simulate the status as "Good" to see if the failure is the only source of trouble.



Parameter	Value
PV	
STATUS	Good_NonCascade::NonSpecific:NotLimited
VALUE	440.000
OUT	
STATUS	Good_NonCascade::NonSpecific:NotLimited
VALUE	440.000
SIMULATE	
SIMULATE_STATUS	Good_NonCascade::NonSpecific:NotLimited
SIMULATE_VALUE	440
TRANSDUCER_STATUS	Good_NonCascade::NonSpecific:NotLimited
TRANSDUCER_VALUE	70.8719
ENABLE_DISABLE	Active
XD_SCALE	
OUT_SCALE	
GRANT_DENY	

Figure 6-11. Fieldbus devices have built-in simulation capability.

EXERCISES

6.1 What is the minimum operating voltage for an IEC 61158-2 device?

6.2 What would likely cause the actual mode of all function blocks to be out-of-service even though the targets have been set?

6.3 What is the maximum distance between the host-end terminator and the interface?

6.4 What is the maximum number of loops per H1 network?

7

Operation

A tremendous amount of information is made available by the fieldbus technologies. They make all process, device, and network information visible from the engineering tool so it can be used by process and instrument engineers. However, to supervise the process the operators only need to access some of this information. Indeed, too much information would only clutter displays and create confusion. Too many pieces of information also slow down the screen-update rate. The process visualization software should therefore be configured to show only the information that is relevant for the operation of the process. When required, more detailed investigation can be carried out using the engineering tool. In other words, the operator normally looks at individual loops, but in the event of failure the technician has access to screens that show the status of devices and function blocks as well as the state of the communications subsystem.

A control loop is usually supervised by accessing information from the PID block in the loop. The programming language of the FOUNDATION Fieldbus control strategy specifies a standard block for PID and other control functions from which the process is also supervised. The procedure for interacting with and interfacing with the block is therefore identical regardless of whether the block is located in a field device or a central controller. PROFIBUS PA and HART do not specify control blocks. Hence, the control is usually done in controllers, using a proprietary programming language that has its own unique parameters for supervising the operation.

Getting the Most Out of Fieldbus

A system may have several operator workstations to make it possible for operation, engineering, and maintenance to be carried out from the same or independent stations. The workstations are connected to the host-level network. The visibility of the plant that is provided from each workstation can be given greater fault tolerance if you use redundant host-level network and linking devices. Even for a small system, it may be a good idea to have at least two workstations so work can continue from one station even if the other fails.

Operators often start to adapt to fieldbus technologies progressively by using the host in the fieldbus system in the same way as they would a traditional system console. Operators then explore the new benefits as they go along. Training can phase operators in one by one.

The process visualization software that the operators interface with should be configured in such a way that it makes full use of the fieldbus features. This will increase the operators' confidence level as well as the ease of use, availability, and safety of the system.

Confidence Level

The operator workstation should display not only the process values, but also the quality of the value and whether there is any limit imposed on it. Once the quality "Good" is displayed the operator will have greater confidence in the value he or she sees. The operator knows whether an abnormal value is a process problem or a measurement problem and need not be in doubt. The status immediately gets the operator's attention if something is not right (figure 7-1). When parameters deviate from their normal value, the operator must be alerted. It is more important to show the operator what is not expected than what is expected. The abnormal information is more valuable.

Safety

Make sure to configure displays to show the quality "Bad" and "Uncertain" adjacent to the values in addition to the limit conditions "High," "Low," and "Constant". This will let the operator spot abnormal conditions and take appropriate action. The block error alarm status from the alarm summary can be displayed alongside the regular process variable alarms. One important safety aspect is that when communication fails the values are displayed as invalid to the operator. A special character such as an asterisk or other reserved symbol will flag non-"Good" OPC values. Numerical values must not be used because they may be mistaken for valid values. Freezing a value that may not be true is dangerous.

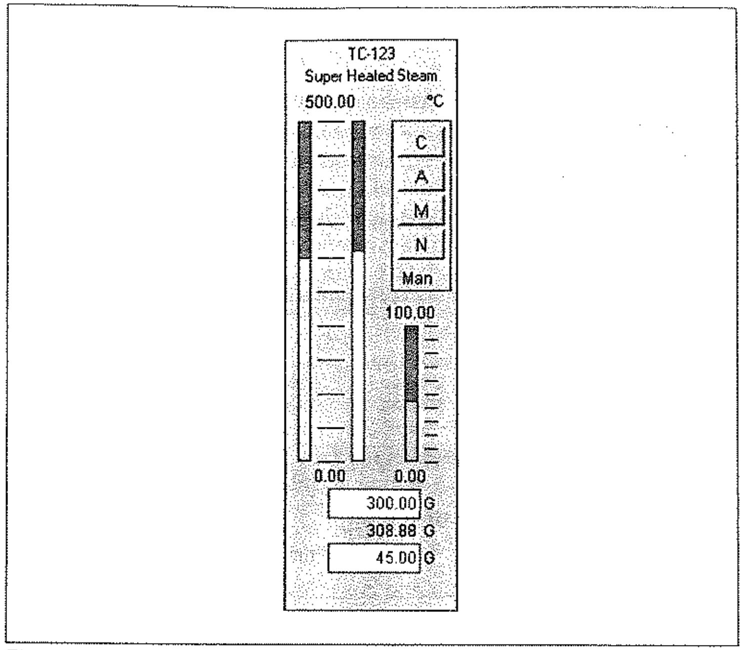


Figure 7-1. Controller faceplate for easy operation. Note the status indication adjacent to the values.

Ease of Use

In addition to the usual Cascade, Auto, and Manual mode buttons it may be helpful for the control loops to include a "Normal" button. The normal mode button sets the Target mode for the loop in the PID block to the mode that is configured in the Normal element of the block mode (MODE_BLK) parameter.

Availability

Status display allows operators to spot problems faster and pinpoint them more precisely. Problems can be fixed sooner so production can back on line. That is, the "mean time to repair" is reduced.

Licensing

One of the main benefits of fieldbus is that more information can be displayed. Traditionally, the process visualization software is sized and priced according to the number of parameters that can be mapped into the database. Such a pricing scheme will be very costly since the number of parameters can easily double. For a data-intense fieldbus system it is more economical to use a licensing scheme in which the number of points communicated at any one time are charged. For example, a system may have thousands of parameters that can be displayed at one time or another, but usually only a few related to one loop detail or to a section of the plant are displayed on the screen. The other parameters are not communicated. In such a licensing scheme the point count is lower, and usually the price is too.

Asset Management

Information about calibration, identification, and materials of construction; detailed diagnostics; and other asset management data—all are too numerous and complex to custom-configure onto screens. For that reason, asset management functions should be carried out from a dedicated asset tool or be integrated into the configuration tool.

Configuring the Process Visualization

The process visualization software displays less information than the configuration tool. However, it displays far more data than was traditionally done by consoles in the past since quality and limit conditions should be displayed adjacent to many of the values. It is a good idea to use process visualization software that can be freely configured to show any parameter from the fieldbus devices. It is important to make use of system templates for faceplates, faceplate groups, and detail tuning screens in order to save display configuration time and to eliminate mistakes. When designing the screens use attractive graphics and make sure that all pages are neat and consistent. Use templates provided by the manufacturer to ensure uniformity. If the standard screen and faceplate templates are unsuitable, it is usually possible to create custom templates.

OPC Data Access

Regardless of the communication protocol a plant uses, the interface hardware usually makes the data available through OPC server software (as explained in chapter 3, "Installation and Commissioning"). OPC technology is using the client/server concept. The hardware manufacturer provides an OPC server for the interface on the host-level network. The software companies provide the OPC client for the software, or, even better, the software is built entirely around an OPC platform. No third party has to be involved, so hardware and software can be freely combined.

OPC client/server technology brings the fieldbus data into the MS Windows environment where the information can be displayed using virtually any process visualization software, since almost all of them support OPC. In other words, the process visualization software is an OPC client. The server software accesses the fieldbus devices using the fieldbus protocol and makes the data available in OPC format. Only a single server is required for all devices on the host-level fieldbus network. The server works with any OPC client software, eliminating the need for custom communication drivers and associated interoperability problems that were common in the past. An OPC client can access data from any OPC server directly without intermediate drivers. OPC thus functions like a software backplane where clients and servers plug in to share data (figure 7-2). The process visualization software usually includes applications for displaying graphics, capturing and logging alarms, performing historical trending and reporting, and so on.

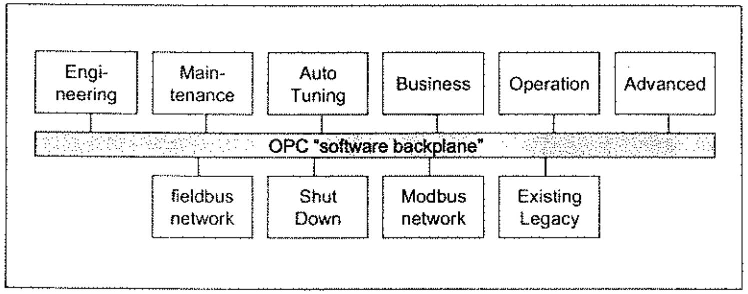
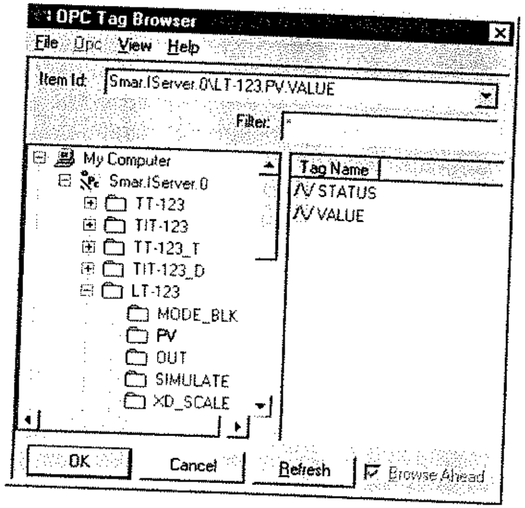


Figure 7-2. OPC is the software equivalent of a backplane.

One limitation of OPC, however, is that it does not handle complex tasks such as configuration download and firmware download, and the like. Therefore, it is important to use a modern standard fieldbus protocol at the host-level network that can handle download and allow other protocols to coexist.

Tag Based

OPC technology is tag based. This makes it particularly easy to use in conjunction with the FOUNDATION Fieldbus, PROFIBUS PA, and HART since they are all based on user-defined tags and standard parameter names such as SP and PV. There is no need to configure memory registers, device addresses, data types, or scaling, among other things. In all OPC client applications the desired data from the server appears automatically and uses the exact same tag that was entered once and for all in the configuration tool. Moreover, every parameter can be picked using an OPC Tag Browser (figure 7-3). For example, when a user-defined tag such as "LT-123" is configured in the engineering tool it also appears in the process visualization software.



Item Id: Smar.IServer.0\LT-123.PV.VALUE

Filter: *

☑ Browse Ahead

Figure 7-3. Point and click to the desired parameter from an OPC Tag Browser.

Single Integrated Database

Usually, the fieldbus configuration tool automatically generates a database for the OPC server that contains all function block tags as well as the parameters in the blocks. No duplicate inputting of tags has to be performed, which saves time and reduces typographical errors. An OPC server becomes the single access point for information from the associated network. All OPC clients throughout the system share access to the different OPC servers. Together, the OPC servers make up a single integrated database, without duplications. All applications utilize the same single integrated database. All OPC clients will show the same values without inconsistencies.

To further integrate applications you can use the function block settings for ranges and limits in the process visualization software to restrict the operator entry range and to scale bar graphs, among other things. Any change from the engineering tool will be reflected in the operator software and vice versa.

Networking

OPC is built on an MS Windows technology called DCOM (Distributed Common Object Model). It was specially developed to bring data across a network, such as to share it among applications in workstations on the Ethernet host-level network. Across the network the information can be disseminated throughout the enterprise. Access to the OPC server can be restricted using standard Windows NT security. A large system may have one server computer that is dedicated for OPC as well as other server applications, whereas in a small system the OPC server application may be running in one of the operator workstations. The OPC client can be a full-fledged operator workstation in the control room or a simpler Windows CE-based local operator LCD panel on the plant floor or in the field. For some systems you will require a high level of fault tolerance to ensure that the operator can see the plant. For these, you can achieve greater availability by using dual OPC server applications running in redundant server stations (figure 7-4). Software at the OPC client side automatically switches to a working OPC server should the primary server fail.

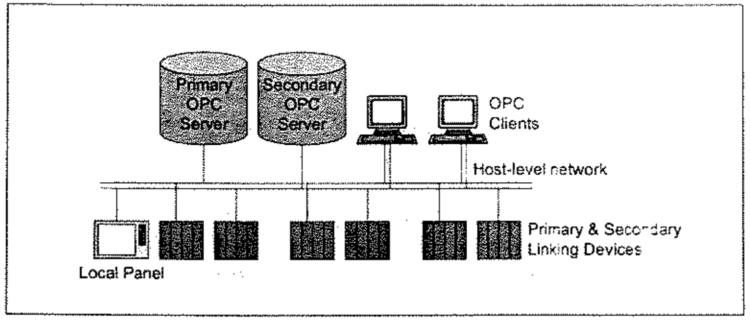


Figure 7-4. OPC server redundancy.

Parameters are accessed throughout the entire network using the same name. OPC offers the additional capability of allowing multiple software applications in the operator workstations, engineering stations, and maintenance stations to access the data at the same time.

Semantics

OPC servers are now available for most hardware and protocols. Thus, it is possible to mix several different protocols in a system and make all the data available in OPC format. However, OPC does not handle the semantics of the data. Different protocols present data in many different formats, using different parameter names and different codes. For example, OPC does not specify how to put a control loop in manual or how to indicate that a flange of a pressure transmitter is made from stainless steel. In one protocol, you may have to write a one to register 10069, whereas in another protocol, device, or application you have to do something else to put the loop in manual.

OPC essentially communicates numbers from the field device to the software applications and vice versa. As a result, an OPC client such as process visualization software must be configured to display the correct meaning of the codes. If several different protocols are mixed in the system, this configuration is still possible, but it becomes more complex. The different protocols do not talk to each other; instead, every protocol has its own OPC server application handling the characteristics of the protocol and interface. An OPC client can access data from any one of them in a standard way, unaware of the underlying protocol. By using an OPC bridge between two servers, data may be passed from one network protocol to another. One important reason to select one single standard protocol like FOUNDATION Fieldbus, PROFIBUS PA, or HART is that these protocols specify the semantics of the data. This makes the meaning consistent throughout all devices, irrespective of manufacturer. This makes it possible to use templates created by the software manufacturer according to the standard parameters, drastically reducing the need for custom configuration. If suitable templates are not available they can usually be created.

Software Interoperability

When you use OPC technology software, your choices are not limited to a single manufacturer. Basically, you can choose any process visualization application that gives you the look and feel that is comfortable for you. Other OPC client applications that can facilitate operation include automatic loop tuning and the linking of data to Microsoft® Excel. A modern control system needs more than just configuration and monitoring software. Software for advanced control, simulation, inferential measurement; statistical process control (SPC), and plant information do not directly support fieldbus protocols. However, these software applications are available as OPC clients, which makes it possible for a wide range of software to be used in a fieldbus-based system. In other words, software applications can be freely selected and interfaced with fieldbus devices.

Because software is playing an ever more important role in plant operation for such tasks as auto-tuning, statistical process control, advanced control and optimization, it is a good idea to use OPC so data is disseminated seamlessly.

Interpreting the Data

Most operation-related values from the fieldbus devices are in floating-point format and in engineering units, and they can be displayed "as is" without any further processing such as scaling. Because there is little or no risk of range inconsistencies the value will always be identical to the value in the field device. However, many parameters in the fieldbus technologies that are relevant to operation are enumerated or bit-enumerated. "Enumerated" means that the parameter contains one piece of information, that every value of the parameter has a particular meaning, and that only one value is possible at any one time. For example, for the FOUNDATION discrete tracking input parameter (TRK_IN_D) a "0" means not tracking, whereas a "1" means that output tracking will be activated. "Bit-enumerated" means that each bit in the parameter has a meaning, and several options can be indicated at any one time.

Typically, each bit corresponds to one piece of information. For example, a feature is either enabled or disabled, or something is true or false. The FOUNDATION and PROFIBUS PA block mode parameter is an example of a bit-enumerated parameter. You will have to configure process visualization software designed to handle any protocol in order to handle the parameter values correctly, either parameter by parameter or using templates. Proprietary operator software may not require such customization.

Status

Many parameters that are essential for operation, such as process variable (PV), setpoint (SP), and output (OUT), have an associated status element in addition to the value. This status is usually shown on line alongside the value in faceplates and in detail screens. The status consists of three parts: quality, limit condition, and substatus. Usually, only the quality and limit parts are displayed in the process visualization software, since the substatus has a level of detail that operators usually don't need. Once a "Bad" or "Uncertain" condition has been detected, engineers or technicians can investigate the cause in detail from the engineering tool. It is therefore a good idea to configure the host to show the status adjacent to the value. The status element used in many parameters is really a combination of enumerated and bit-enumerated.

Quality Status

The two most significant bits of the status byte indicate the quality. For that reason, you should do a bitwise AND with 0x11000000 (192 decimal) in order to mask out the limit and substatus bits. Depending on the result of the masking, the quality can be displayed as an enumerated state with a value and state string for the four qualities (table 7-1).

Table 7-1. Quality enumeration.

Value	Display	Description
0	B	Bad
64	U	Uncertain
128	G	Good (non-cascade)
192	G	Good (cascade)

To reduce clutter on the display consider not displaying any text in case the quality is "Good."

Limit Status

The two least significant bits of the status byte indicate the limit condition. You should therefore do a bit-wise AND with 0x00000011 (3 decimal) in order to mask out the quality and substatus bits. Depending on the result of the masking, the limit condition can be displayed as an enumerated state with a value and state string for the four limit conditions (table 7-2).

Table 7-2. Limit condition enumeration.

Value	Display	Description
0		No limit
1	L	Low
2	H	High
3	C	Constant

Block Mode

The block mode parameter consists of four elements: Target, Actual, Permitted, and Normal. The two most important ones for plant operation are Actual and Target. In a control loop built using the FOUNDATION Fieldbus programming language the mode of the control loop is usually set in the PID function block. The mode of other blocks is usually not used. Writing to the Target element of the block mode parameter sets the desired operating mode for the control loop. The present mode of the loop is displayed by reading the Actual element of the block mode parameter.

There are many ways to design the user interface for the mode of the loop. For the FOUNDATION Fieldbus programming language, the most straightforward way is to use three buttons for the Target modes and a separate indication of Actual mode.

Set Target Mode

To set Target Mode, use three buttons, for example, one each for Manual, Automatic, and Cascade. Set the mode by writing the corresponding code to the Target element of the block mode (MODE_BLK) parameter (Table 7-3).

Table 7-3. Target mode coding.

Caption	Value	Description
M	16	Manual
A	8	Automatic
C	12	Cascade

It is even possible to make a fourth button in order to set the "normal" operating mode of the loop, provided the button has been configured in the block at the Normal element of the block mode parameter. When the normal mode button is pressed, or perhaps activated through some script for batch control, simply write the value at the Normal element to the Target element. This feature may prove a useful feature since it provides a simple way to return a loop to normal mode after service or load change, when the operator may have forgotten which mode the loop was in originally.

The device itself checks that the requested mode is enabled in the Permitted element of the block mode parameter.

Display Actual

The actual mode will be read from the Actual element of the block mode (MODE_BLK) parameter. The mode can be displayed as an enumerated state with a value and state string for the eight modes (table 7-4).

Table 7-4. Actual mode enumeration.

Value	Display	Description
128	OOS	Out of Service
64	IMan	Initialization Manual
32	LO	Local Override
16	Man	Manual
8	Auto	Automatic
4	Cas	Cascade
2	RCas	Remote Cascade
1	ROut	Remote Output

Alarm Summary

There are many ways to display alarm status for an individual loop in a faceplate or loop detail screen besides a dedicated systemwide alarm screen. Because the alarm summary parameter is bit-enumerated the alarms should be filtered out one by one to obtain independent indicators for each alarm. As a result, you should do a bitwise AND with the corresponding mask. If the unmasked bit is true, the corresponding mnemonic should be displayed (table 7-5).

Table 7-5. Alarm enumeration and masking values.

Value	Display	Description
128	B	Block
64	DL	Deviation Low
32	DH	Deviation High
16	L	Low
8	LL	Low-Low
4	H	High
2	HH	High-High
1	D	Discrete

To reduce clutter on the display it may be a good idea to avoid displaying any text in case the alarm is not active.

Using Scaling Information

A lot of information is stored in the function blocks, for example, scaling and limit information. You can use the scaling information when you need to create animation for the value, for example, animation of a faceplate bar graph or animation to mimic the tank level of the actual process.

The scaling of the bar graph in the process visualization software must match the scaling of the device in order for the bar graph indication to correctly represent the percentage value. To eliminate the chance of any inconsistencies the range values of the bar graph should access the scaling value from the field device. In this case, any change that occurs to the range in the field device is taken into account when the bar graph is displayed, that is, a single integrated database located in the field device is achieved where a range change in the configuration tool automatically updates the process visualization software. Use the EU_0 and EU_100 elements of the appropriate scale parameter for 0 percent and 100 percent, respectively. The related scale parameters are as follows:

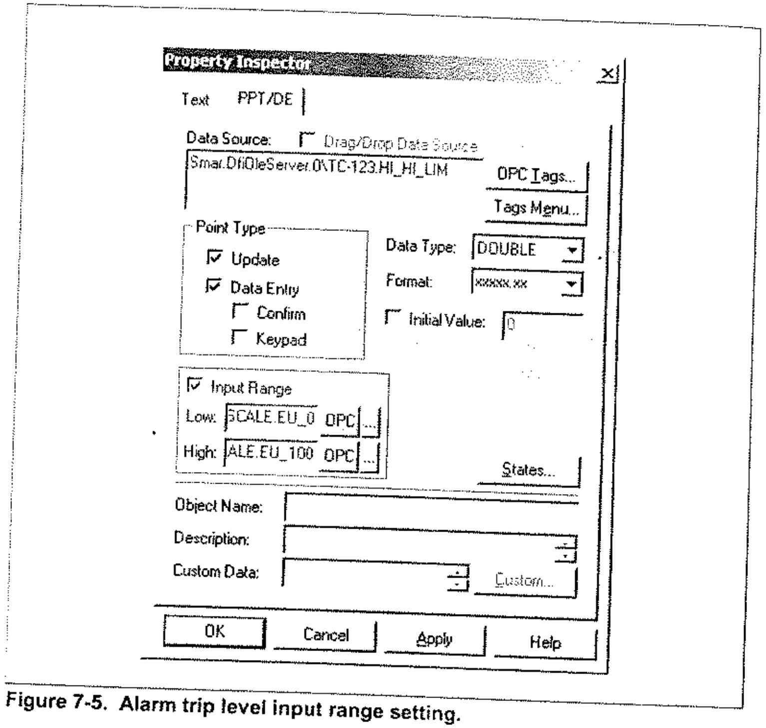
PV and SP:	PV_SCALE
OUT:	OUT_SCALE

It may be effective to also display the scaling end points as values above and below the faceplate bar graph.

It is not essential to check alarm trip level values at operator entry because the field device itself prevents the written value from exceeding the configured limits. However, it desirable to have the process visualization application inform the operator of any limit violation and to indicate the valid range. Therefore, it is a good idea to configure the data entry limits for the alarm levels in the process visualization software so they follow the process variable scale. In other words, the process visualization software should access the scaling parameter in the device just as it does for the bar graph end points (figure 7-5).

Using Limit Information

It isn't essential to check setpoint and output values at entry because the field device itself keeps the written value within the


Property Inspector

Data Source: ☐ Drag/Drop Data Source
Smar.DfiOleServer.0\TC-123.HI_HI_LIM

Point Type
☑ Update
☑ Data Entry
☐ Confirm
☐ Keypad

Data Type: DOUBLE
Format: xxxxx.xx
☐ Initial Value: 0

☑ Input Range
Low: SCALE.EU_0
High: ALE.EU_100

Object Name:
Description:
Custom Data:

Figure 7-5. Alarm trip level input range setting.

configured limits. However, you do want the process visualization software to inform the operator of any limit violation and to indicate the valid range. For certain parameters, for example, SP and OUT, some function blocks in the FOUNDATION Fieldbus programming language have dedicated limit parameters. These are as follows:

For setpoint:

High:	SP_HI_LIM
Low:	SP_LO_LIM

For output:

High:	OUT_HI_LIM
Low:	OUT_LO_LIM

The user entry limits that are set in the process visualization software should match the limits in the field device. To eliminate the chance of any inconsistencies, the process visualization software can access these limit values from the field device. In this case, the software takes into account any change to the limits in the field device when the operator enters a value, that is, a single integrated database where data is stored in the field instrument and the same value is used both for alarm detection and operator display and where a change from the configuration tool automatically updated the operator display too.

Homogeneous Operation Environment

FOUNDATION H1 and HSE devices use priority levels 0-15 for alarms and events, as described in chapter 4, "configuration." For consistency, these same levels should be used throughout the system in order to eliminate operator mistakes. In other words, alarms detected in any shared central controller or in software should use the same levels.

Performance Considerations

The more information is displayed on the screen, the more information there is that must be communicated. A screen with a great deal of information is not only cluttered; the update will also be slower than for a leaner screen.

It is a good idea to make sure that the host is using View objects or Multi-Variable Container (MVC) objects because these two objects allow the host to access multiple pieces of information from the fieldbus devices in a single, efficient communication rather than in multiple requests.

Some steps can be taken to minimize the screen refresh time.

Demand Based

You should use the kind of process visualization software that does not scan parameters that are not being accessed. This will prevent the OPC server and network from being loaded unnecessarily. In other words, the software will work on a demand scan basis, only accessing information that is being displayed, alarmed, trended, or used otherwise.

Reduce Views

By reducing the number of view objects that have to be accessed to display a screen you can reduce the communication needed to retrieve the information. It is therefore a good idea to restrict the information shown in a "faceplate" type display to the parameters in VIEW_1 and VIEW_2. A "detail screen" can also display the parameters in VIEW_4. To gain additional information from the block it is better to use the configuration tool.

Reduce Scanned Blocks

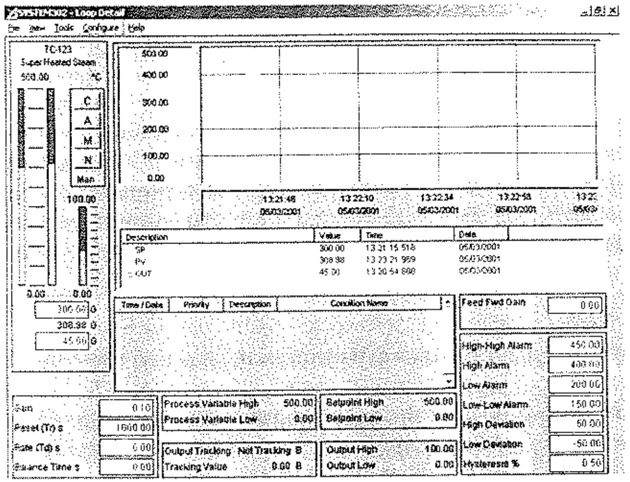
Another important measure for improving performance is to reduce the number of blocks that are scanned by the host for supervisory functions. Limit the continuous data access for normal operations to just one block in the loop, typically the PID block. If variables are repeatedly gathered from the AI, PID, and AO block the communication load becomes three times as heavy. The AI and AO blocks have the status options (STATUS_OPTS) "propagate failure forward" and "propagate failure backward," respectively, which ensure that the relevant information ends up in the PID block.

Multi-variable Container

An MVC (Multi-Variable Container) is a feature found in some FOUNDATION devices. The host configures the MVC to access data from several different blocks in a single communication, thereby optimizing communication. The MVC is ideal for converter products that have many I/O channels and function blocks, which otherwise would require that many view objects be communicated.

Operating a Fieldbus Control Loop

Most process visualization software applications allow screens to be totally customized. It is very common to configure screens similar to those found in traditional DCS. That is, in addition to process flow mimic, alarm and event logs, and historical trending screens there may be overview screens, faceplate group screens, loop tuning detail screen, and so on (figure 7-6). These are often arranged in a hierarchical manner, following the ANSI/ISA-S88.01-1995 model, which represents the arrangement of physical plant equipment. Both the faceplate and loop-detail screens for tuning contain a great deal of information accessed from the fieldbus device.



TC-123
Super Heated Steam
500.00 °C

306.00 G
308.98 G
45.00 G

Description	Value	Time	Date
SP	300.00	13:31:15.518	05/03/2001
PV	308.98	13:23:21.959	05/03/2001
OUT	45.00	13:30:54.898	05/03/2001

Gain 0.10
Reset (Ti) s 1600.00
Rate (Td) s 0.00
Balance Time s 0.00

Process Variable High 500.00
Process Variable Low 0.00

Setpoint High 500.00
Setpoint Low 0.00

Output Tracking Not Tracking B
Tracking Value 0.00 B

Output High 100.00
Output Low 0.00

Feed Fwd Gain 0.00

High-High Alarm 450.00
High Alarm 400.00
Low Alarm 200.00
Low-Low Alarm 150.00
High Deviation 50.00
Low Deviation -50.00
Hysteresis % 0.50

Figure 7-6. Loop detail screen for tuning, etc.

Because a large amount of information has to be configured on each screen, it is a good idea to use preconfigured templates to save time.

Many parameters are read-only, for example, PV. These read-only parameters cannot be changed. When the system is on line some parameters can only be written in certain modes, for example, OUT only in Manual, SP only in Auto or Manual, and so on.

Operation

Basic operation of the control loop includes monitoring the process variable and changing the mode, setpoint, and output to initiate control actions. If the Actual mode does not match the set Target mode, this is an indication that something in the loop is not normal, though not necessarily a failure. For example, if the Actual mode for the primary PID in a cascade is IMan, this may simply indicate that the secondary has been taken out of Cascade mode. If the mode is Local Override, this indicates that the output tracking is active, which may be perfectly all right. Similarly, in some circumstances a demand for a Target mode change, such as from Manual to Automatic, may not be executed unless everything in the block is OK.

The process variable value is typically displayed together with the quality and limit condition. Along with the numerical value, there is usually a bar graph scaled according to the process variable scale values. Similarly, the setpoint is displayed with quality and limit conditions together with a bar graph that uses the same scale. Additionally, the setpoint may be restricted by more stringent limits. The setpoint of the secondary controller cannot be written in cascade mode. The output is also shown with quality and limit conditions together with a bar graph scaled according to the output scale parameter in the device. Additionally, the output may be restricted by more stringent limits. The output cannot be written in automatic or cascade mode.

Tuning

The loop response for a PID controller function block is tuned using the parameters Gain, Reset, and Rate. Gain is a dimensionless parameter, whereas Reset and Rate are configured in seconds. Integral action is canceled by configuring the Reset parameter infinitely high (+Inf). Derivative action is canceled by configuring the Rate parameter to zero. If the feedforward function is used, you can tune the contribution to the output from the Feedforward gain parameter. See, for example, the book *Fundamentals of Process Control Theory* by Paul W. Murrill for the details on how to actually select the tuning parameter values (among other ISA publications).

Alarms

Generally, the operators are allowed to set alarm limits too. These limits have to be set within the range determined by the process variable scale and in the same units. Alarm hysteresis is configured in percentage terms.

EXERCISES

7.1 Does status differentiate between process problems and device problems?

7.2 Can multiple options be active in an enumerated parameter?

7.3 Does it make a difference for the display of updated time if the same number of parameters are accessed from different blocks or from a single block?

7.4 Can the output (OUT) parameter be written in IMan mode?

BIBLIOGRAPHY

1. Murrill, Paul W., *Fundamentals of Process Control Theory*, 3 d ed. Research Triangle Park, NC: ISA, 2000, ISBN 1-55617-297-4.

8

Engineering and Design

One of the many reasons to adopt a fieldbus technology is that it simplifies the implementation phase of a project, thus reducing both cost and time. Though fieldbus technologies change the architecture and capabilities of process control, the basic project cycle and methodology for system engineering and design remain the same as before fieldbus. Inevitably, these fieldbus technologies necessitate some new engineering practices. At each phase of the system implementation there are a number of small changes, such as different considerations, different documentation, and so on. However, tools have become available for handling these new practices. The design work must be based on the capabilities and limitations of the fieldbus technologies discussed in the chapters on installation (chapter 3) and configuration (chapter 4). It is therefore important that you have read and understood these two chapters fully before embarking on this chapter.

HART is almost exclusively used in single-unit mode, thus utilizing the 4-20 mA signal. Therefore, the engineering practices for HART are no different from those of conventional systems, which are already well understood. A system should accept—in addition to a fieldbus technology—conventional analog and discrete I/O engineered in the same old way as before.

Many companies have internal standards and procedures for the installation, engineering, and specifications of the equipment to be used. Before you embark on a fieldbus project, you should review and update these documents so they reflect the new requirements and possibilities enabled by the fieldbus technology you have chosen. Choose a small segment of the plant as a pilot project to evaluate standards and documentation and get hands-on experience.

Conceptual Design and Functional Specification

The first step is to define the requirements that the process is demanding of the control system: how many transmitters and final control elements are required and so on. Once you know the number of devices needed you can work out the initial system definition for how these devices shall be networked together and interfaced to the operators. Keep in mind that many fieldbus devices have multiple channels and therefore a single instrument can be used where previously separate transmitters were required. Similarly, a fieldbus valve positioner typically has software limit switches, which reduce the need for discrete inputs and components in the field.

To gain an initial estimate of the communication port count it may be a good idea to work with the assumption that there will be twelve devices per network. This leaves room to expand up to sixteen in the future. It is not necessary to know if devices are analog or discrete, nor if they are input or output, because the fieldbus signal is the same for all. That is, for fieldbus you count the number of devices, not the number of "points," as in conventional systems. Since field devices usually do the control there is no need to differentiate between control points and measurement points. Based on the port count, estimate the number of linking devices or interface modules that are required. Determine which of the field-level networks will require fault-tolerance measures such as redundant interfaces and the like.

Do some preliminary checking to determine if all the required device types are available for the particular fieldbus technology you intend to use. For a modern system, it may be a good idea to use a low-cost fieldbus pressure transmitter where a pressure switch would have been used in a conventional system. Next, determine what parts of the system are fieldbus and what are conventional. Ideally, you will eliminate as much conventional I/O as possible and find a fieldbus alternative instead. However, your system most likely will also require conventional I/O. Decide if these I/O shall be located on the field-level network using field-mounted converters or on an I/O subsystem on the host-level network in panels. Make sure to deduct any conventional I/O that was changed into fieldbus from the conventional I/O point listing.

When developing a functional specification for the loops make sure to use fieldbus terminology.

Engineering

Detail system design includes specifying and selecting devices, configuring control strategy, and creating documentation. For the first project, it is common for the engineering work to be overelaborated with in-depth circuit analysis, and the like. As the project progresses, it becomes clear that the technologies and devices are not particularly restrictive and therefore that, unless extreme, networks usually fall within limits.

System and Control Strategy Design

Offline configuration means the ability to configure the system without having to be connected to any device. Being off line makes it possible for the system to be configured immediately, without having to wait until the system is moved on site and every instrument has arrived. It is therefore a good idea to use a fieldbus tool that supports offline configuration. For FOUNDATION Fieldbus this means that the configuration tool must support the standard capabilities files.

Devices and Networks

Next, generate the bill of materials in accordance with the requirement at each hierarchical level of the architecture.

Once you have determined the number of field-level networks based on device loading you can calculate the number of network ports needed and the field power supply requirement. On average, devices consume less than 20 mA each, most devices less than that. Hence, by selecting power supply impedance with an output of up to 320 mA per port you will have sufficient power and most likely lots to spare. Thus, you can eliminate the need to make a power budget for each network. Decide which field-level networks require redundant field power. By choosing devices with low power consumption, 12 mA or so, you reduce cable limitations caused by voltage drop, eliminating the need for detailed voltage drop analysis.

Based on the topology you desire and the cable type, select the installation accessories, such as junction boxes. Try to stick to one device per spur in order to enjoy maximum spur length. Do not exceed sixteen devices per network for most applications, but aim for less than twelve for the initial system to allow for expansion. As far as possible, locate all devices that participate in the same loop on the same network. This will improve performance by avoiding the need to use bridge functionality between networks. Using good cable, the IEC 61158-2 wiring can run for a very long distance, far exceeding what is normal in most plants. For most networks, there is therefore no need to calculate the length and voltage drop at each device since it is usually much shorter than the maximum allowed. However, if there is some network that runs to a remote unit such as waste treatment or nitrogen production the wire run may be longer than it would be within a building. Calculating the length in this case may be a good idea. For intrinsically safe networks, you should complete an entity parameter sheet for each intrinsically safe segment. If you choose the FISCO concept for intrinsic safety the engineering work is reduced in comparison to the traditional entity concept.

It may be very elegant to dedicate linking devices so they are not shared between process cells. However, if you do, more linking devices may be required, which will increase cost. As in many other cases, strike the right balance between simplicity and economy. For simplicity, it may also be a good idea to use a linking device with built-in field device power. Decide which field-level networks require redundant interfaces to ensure that there is adequate availability for the communication.

For the host-level network, decide if redundancy is required and if there are any long stretches where fiber optics is necessary. Depending on the system size consider using some switching hubs or even routers.

Control Strategy

When designing the control strategy make sure you make full use of the status capability of both FOUNDATION Fieldbus and PROFIBUS PA. In other words, use the status bytes throughout the control strategy to provide fail-safe action, reset windup protection, and bumpless transfer. In a system that uses the FOUNDATION Fieldbus protocol this is easiest done simply by using the FOUNDATION function blocks in all devices without involving proprietary languages, including any central controllers. For PROFIBUS PA, make sure you select a central controller that supports the use of the status bytes, preferably as an integral part of the language.

For FOUNDATION Fieldbus, select blocks, link them, and parameterize. Then allocate function blocks in the field devices. Avoid centralizing blocks in a central controller as this makes the system more vulnerable.

Deep inside the plumbing of the FOUNDATION Fieldbus protocol there is something called a VCR (Virtual Communication Relationship). VCRs are difficult for people without expertise in fieldbus to understand, and it is therefore important that the complexity of VCRs be hidden from the user to make configuration easy. For you propeller heads, VCRs are explained in chapter 11, "How Fieldbus Works." In each device, one VCR is required for each communicated link as well as additional ones for other communication functions. Function blocks should be allocated into devices in such a way as to reduce the number of communicated links. This will improve control performance, with the added benefit that the number of VCRs required is reduced. For a device that supports very few VCRs, it may be prudent at the time of engineering to check that it has sufficient VCRs for the application. Devices that have few function blocks and few VCRs can be constrictive, which makes the allocation of blocks to devices an iterative process. It is a good idea to use devices that support a large number of VCRs. Doing so, eliminates the need to calculate a VCR requirement budget, which is simply too much work. As a last instance, the configuration tool checks that the number of VCRs has not been exceeded. Therefore, the sooner the configuration entry starts, the better. Waiting until all devices arrive on site may be too late.

When you design the control strategy work as far as possible with the standard FOUNDATION Fieldbus blocks. This will make it easier to replace the device in the future without having to replace any manufacturer-specific blocks, and it gives you greater flexibility to assign the blocks for execution in different devices. In case the standard blocks don't suffice, use extended or specific blocks. It is a good idea to use a configuration tool that supports profiles for identifying blocks. With profile support it is possible to build a control strategy that is completely independent of the particular device that will eventually be used. That is, the process engineers can build the control strategy without really knowing the devices and can let the instrument engineers assign the blocks to devices without really caring about the strategy.

Operator Interface

Select the number of workstations required. Even for a small system, it may be a good idea to use at least two to provide redundancy. In a small system, the operation, engineering, and maintenance can be carried out from a single station, but in a larger system these functions may have dedicated stations. In a small system, server for OPC data access, alarms and events, and historical trending can be in one workstation. However, for larger systems such functions are best done in a separate server.

Selecting Vendors and Parts

The process of selecting the host, field device, and software is fairly straightforward because devices from different manufacturers are interoperable. The painful process of selecting just a single vendor is therefore eliminated. In other words, you don't have to make the difficult compromise of having to choose either the best transmitter, the best valve, or the best software, and so on. Although fieldbus technologies are standards, there is plenty of differentiation in the capability devices offer, in diagnostic coverage, and so on. The required functionality must be specified.

The differences between fieldbus and conventional devices should not be overdramatized. Engineering and specification is not very different since the basic principle of measurement remains the same. The device specification sheets should just say what type of fieldbus it shall be in place of the usual 4-20 mA signal. The rest is the same. The specification sheet does not have to include features that are implicit to the fieldbus technology chosen. For example, for a controlling device it is only necessary to specify that it shall use the FOUNDATION Fieldbus PID function block, without having to go into details about reset windup protection, bumpless transfer and safety, and so on. This means that system specifications become simplified. Similarly, don't define detailed requirements repeatedly for each loop or device tag. Instead, specify the common requirements for each device type and apply them consistently throughout the entire plant. The engineering burden is increased if you use devices with many restrictions such as high current consumption, limited or fixed set of function blocks, few VCRs, and the like. If very capable devices are used much work can be eliminated.

Field Instrument Specifications

If you used devices with a fixed or limited set of function blocks, selecting the transmitters and positioners appropriate for the application may be an iterative process. For example, a device chosen for the application in the first round may be found to lack a sufficient number of blocks. This may force you to choose another device, which in turn has to be checked against the process requirements. Some devices have dynamically instantiating function blocks, meaning that they have a library of many block types and many copies of each block type can be used in the device. You should use field devices that have dynamically instantiable function blocks to eliminate this constraint, thereby making the engineering process straightforward. Dynamically instantiable blocks give you maximum flexibility when building the control strategy.

The number of devices you can have on a network segment and the length of the network segment both depend on the voltage drop along the wire. By choosing devices with low power consumption these constraints greatly diminish. Consequently, for most of the networks in the plant there is no need to specifically calculate whether the network will operate within the constraints. Low power consumption has the added advantage of enabling more devices to be connected per safety barrier in intrinsically safe installations.

The existing plant standard specification for transmitters, positioners, and other field instruments can be used with an added clause containing some points for fieldbus requirements. Essentially, the existing ISA-20-1981 specification sheets can be used "as is." Just indicate that the signal shall be FOUNDATION Fieldbus H1, PROFIBUS PA, or 4-20 mA with HART. The individual data sheet for each tag should not go into details about the device type specification. You should put the performance and feature characteristics that are common to all the required devices of a particular device type into a separate specification. Examples of points to include in separate detail specifications are the following:

- *Required function block types: PID, Arithmetic, Signal Characterizer, Integrator, Analog Alarm, Input Selector, Timer, and Lead-Lag.
- *Dynamically instantiable blocks: Yes
- *Maximum execution time for each block type: 70 ms for the blocks just mentioned
- *Block quantity: more than 12
- Bus power consumption: 12 mA or less
- Intrinsic safety: Exia IIC, FISCO
- *VCR: more than 40
- *Link master capability
- Registered as interoperable

Note: the points indicated by asterisks are only applicable to FOUNDATION Fieldbus devices.

Host Device Specification

The host includes interface hardware such as a linking device and possibly a central controller. Also required are operator and engineering workstations with their respective software as well as a host-level network. For FOUNDATION Fieldbus hosts it may be a good idea to review the Host Interoperability Support Test (HIST) report to confirm the capabilities.

Interface Hardware Specification

To achieve high availability, you should use a redundant host-level network and interface modules so as to ensure two paths to the operator. To simplify both engineering and construction, the fieldbus device power supply subsystem should be integrated with the interface in a single device (figure 3-28). To simplify commissioning, eliminate human error, and reduce the number of cross-reference list documents the assignment of addresses for the field-level network shall be automatic.

You can use the existing plant standard specification for the host hardware with an added clause containing some points for fieldbus requirements. Indicate that the host-level protocol shall be FOUNDATION Fieldbus HSE or PROFIBUS PROFInet. The following are some examples of points to include in a detail specification for a linking device:

- Redundant field-level network interface (in different backplanes)
- Redundant host-level networks
- *VCR: more than 50 per port
- *Automatic address assignment on field-level network
- OPC server data access version 2.0 compliance-tested
- Integrated redundant power supply and impedance (in different backplanes)
- Field-level ports: minimum 4
- *Publisher/subscriber communication of process I/O
- *FOUNDATION Fieldbus language function block execution: PID, Arithmetic, Signal Characterizer, Integrator, Analog Alarm, Input Selector, Timer, and Lead-Lag

- *Conventional I/O represented by regular input and output function blocks: Analog Input, Discrete Input, Analog Output, and Discrete Output.

Note: the points with an asterisk are only applicable to FOUNDATION Fieldbus devices.

Engineering Software Specification

Building a fieldbus system is not as simple as installing a fieldbus interface like another module in the I/O of an old system. The host must also have the software to support all the new functionality; it cannot just use the process I/O. For a small pilot system, a stand-alone fieldbus configuration tool may work. However, for a full-fledged system the best solution is an integrated engineering tool for fieldbus device, control strategy, and network configuration. The reason for this is that if the control strategy and device configurations are made in different tools with different databases a lot of extra work will be created, raising the risk of inconsistencies and thus a nonfunctioning strategy. For example, suppose the range is configured as 0-200 in the device but 0-500 in the controller or operating screen. The alarms will not trip as they should, and controls will not work, so operators will be misled.

On the other hand, errors are eliminated if all the configuration is done from a single tool with a single database, so that the data is entered only once in a single place. This is particularly true when future changes need to be made because the risk that personnel will forget to enter data in a second or third place is greater. To benefit from asset management, you should design the system so the tool is permanently on line. This will facilitate easy access to the live list, maintenance, and calibration information when the plant is operating.

For the host to be truly interoperable and able to configure any device regardless of manufacturer it should only require the standard device support files associated with the protocol used. If the host requires additional device support files from the host manufacturer, no independent institutes will test these files to "certify" each version of every device made. The host manufacturer will consequently have to test these files to make sure the proprietary files are OK. The use of proprietary files reduces interoperability to just a small shortlist of devices.

Network drawings and P&I diagrams show devices and function blocks, respectively, but not how blocks are allocated to devices.

The configuration tool should therefore automatically generate such a cross-reference list. Other useful engineering documents are those illustrating block parameterization and the allocation of devices to the networks (figure 8-10).

Field instruments and other devices used in large quantities are not off-the-shelf items. Therefore, to minimize engineering time the configuration must begin long before these devices have been manufactured and installed. The engineering tool should therefore be able to build control strategy and to configure devices off line without any field instruments or controllers being connected. Because FOUNDATION Fieldbus and PROFIBUS PA have function blocks in the field devices the boundary between configuring devices and configuring control strategy becomes blurred. It's a good idea to use a configuration tool that allows the control strategy to be configured independently of the device where it will be executed. The block can be allocated to devices later, that is, the device issues and control strategy can be kept separate.

You can use the existing plant standard specification for the engineering software by adding a clause containing some points for fieldbus requirements. The following list gives examples of useful features and characteristics explained in the earlier chapters that you should consider including in a detail specification:

- *No proprietary device support files should be required, just standard device description and capabilities files.
- Automatic generation of documentation
- Display of live list
- Offline configuration
- Dynamic block instantiation
- Permanent online access to instruments
- Auto-generate OPC tag list
- Support of profiles in order to identify block capability

Process Visualization Software Specification

A prerequisite for easy engineering is that there is no duplication of databases. Therefore, it is a good idea to use process visualization software in the operator consoles that accesses data directly from the OPC server without needing to map into intermediate databases.

You can use the existing plant standard specification for the process visualization software with an added clause containing some points for fieldbus and OPC. The following are examples of useful features and characteristics explained in the earlier chapters that you should consider including in a detail specification:

- OPC client
- Direct OPC access
- Non-Good OPC values are flagged by a special character such as an asterisk or other reserved symbol. Numerical values must not be used.
- Display quality and limit conditions next to values
- *Use FOUNDATION alarm priority levels throughout the system
- Support the 32-character TAG_DESC, function block tag, physical device tag, and parameter names used in FOUNDATION and PROFIBUS PA.

Package Units

The basic control system usually has to be integratable with packaged units for boiler control, compressor controls, and other functions around the plant. Package units are equipment that come with their own preinstalled instruments and controls. To simplify engineering and integration work make sure to specify that all package units must have a connection for the same host-level network utilized by the basic control system.

System Configuration Entry

Typically, the control strategy, network architecture, and field device configurations are built in the offline mode from the configuration tool once the control strategy and device requirements have been decided. This work can be completed before the devices have been manufactured or connected. Downloading is done during the commissioning stage. Generally, this configuration is made from a single tool that allows function blocks to be assigned to devices and devices to be assigned to networks by simply dragging and dropping their icons. Before you begin configuration you must load the device support files for the relevant device types on the computer unless they were preinstalled. By starting configuration when the project is launched, you can reduce the engineering and start-up time. The configuration of device and control strategy is prepared in advance from any location. The configuration is stored as a file, brought on site, and downloaded once the devices are in place. The engineering tool continuously checks the configuration's validity to prevent users from making illegal configurations. It also verifies the configuration against the device's resource file to ensure that no more resources are allocated than the device supports.

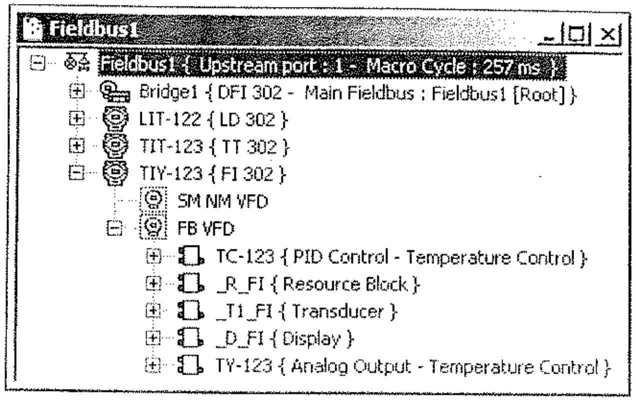
Because FOUNDATION Fieldbus is a communication protocol and programming language for configuring control strategies, devices and control strategy can be configured using the same tool. The engineering tool schedules the communication so it is synchronized with the block execution.

Devices and Network Configuration

Configuring field-level networks is essentially a matter of assigning the networks to the respective communication ports and enabling backup link masters. Using templates for devices speeds up the configuration. For FOUNDATION Fieldbus and Profibus PA these templates would include the configuration for the resource (physical) block and for the transducer blocks in the device. If standard templates are unavailable, they can usually be created. It is a good idea to tag all the networks in the plant. You may simplify future maintenance of the network by naming the networks around the plant according to some logical tagging scheme similar to that used for any wiring. It may be particularly helpful if the network tag is also stored in the configuration tool. The network tag should follow the name it was given in the network drawing.

Control Strategy Configuration

You can execute FOUNDATION Fieldbus function block language anywhere in a transmitter, positioner, or central controller. Therefore, it is possible, and necessary, to select where each block shall be executed. Generally, the configuration tool shows the control module name for the loop that the function blocks in a device are part of (figure 8-1).



```
Fieldbus1 { Upstream port : 1 -  Macro Cycle : 257 ms  }
    Bridge1 { DFI 302 -  Main Fieldbus : Fieldbus1 [Root] }
    LIT-122 { LD 302 }
    TIT-123 { TT 302 }
    TIY-123 { FI 302 }
        SM NM VFD
        FB VFD
            TC-123 { PID Control - Temperature Control }
            _R_FI { Resource Block }
            _T1_FI { Transducer }
            _D_FI { Display }
            TY-123 { Analog Output - Temperature Control }
```

Figure 8-1. The loop origin displayed adjacent to the function block.

Typically, the system configuration begins by building the control strategy, then selecting the devices, and then allocating the blocks. Typically, the regulatory aspects of the control strategy are handled first. The various options for control, status, and I/O are handled later in order to achieve the desired behavior and maximize safety without sacrificing availability. The configuration tool shows the execution time for the H1 network based on the blocks selected and the links made. A long macrocycle, that is, the time it takes to execute the function blocks and communicate their links, is an indication of an overloaded network. The sooner the configuration starts, the sooner the limitations of the devices, if any, can be discovered. It is therefore good to start configuration entry early off line.

When function blocks are allocated to a device, the configuration tool continuously checks that the device's block execution, link communication resources, and the like as specified by the capabilities file are not exceeded. If the device does not support a particular block type the engineer is notified. Ideally, the devices support dynamic instantiation of function blocks, which provides greater flexibility when assigning the blocks. It may be beneficial to adopt a rule for allocating blocks such that there are no output block links from final control elements other than feedback. This way, you can remove final control elements without affecting other loops. Do not allocate many blocks to one device since centralizing functionality increases vulnerability. Any device that has blocks allocated from more than a few loops should have redundancy to ensure adequate availability.

Most configuration tools provide a library of preconfigured and pretested strategy templates for common control schemes. Each template includes linked function blocks and set parameters. Templates facilitate neat and consistent configuration with fewer errors. They also simplify the verification, as it becomes easy to prove the correctness of the configuration.

It is a good idea to use a single configuration tool that treats fieldbus and conventional I/O the same way. This will make it easier to implement a hybrid configuration consistently.

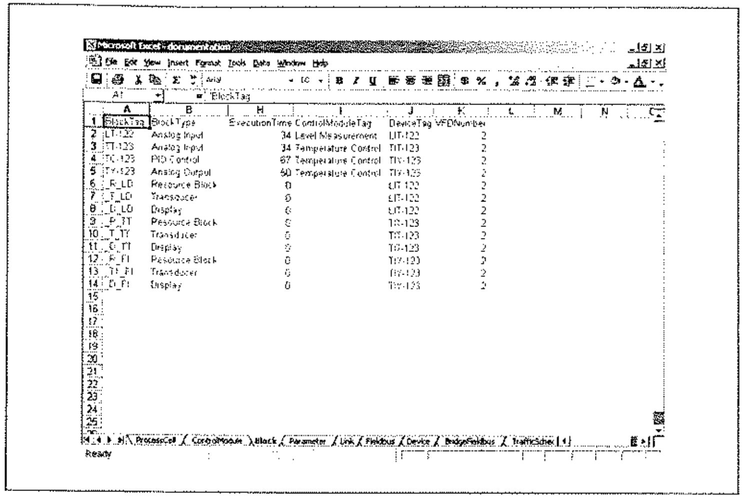
Operator Interface Configuration

The operator interface usually lets the operator supervise the loop by accessing the controlling device. This is beyond the scope of PROFIBUS PA and HART, but it is part of the FOUNDATION Fieldbus programming language. Once a function block has been instantiated in the configuration application the function block with all its parameters becomes available to all other software applications. For example, the OPC clients can access all parameters using the universal tag browser. Any graphics that have been created can then be linked to their respective function blocks. In many systems using OPC this is possible even without configuring an intermediate database.

Documentation

The documentation process for a fieldbus system requires only slight alterations from past practices. Usually, the configuration tool automatically generates part of the documentation as the configuration is entered, for example, lists of networks, instruments, blocks and parameters, and the like. These listings make ideal checklists to use when doing FAT (factory acceptance test) and SAT (site acceptance test) verification.

When several devices are multidropped it is more suitable to use a network diagram with all devices on the us than the traditional loop diagram. In FOUNDATION Fieldbus and PROFIBUS PA, the boundary between devices and control strategy can become confused. To keep the design process simple, do not let the P&I diagram and network diagram, which replace the loop diagrams, have overlapping information because duplicated data results in inconstancies. The physical wiring is best shown only in the network diagram and the control scheme functionality only in the P&ID, just like in the past. Devices and the physical device tag (PD_TAG) are best shown in the network diagram. Function blocks and function block tags (FB_TAG) are most suitable for the P&ID. The configuration tool usually generates a cross-reference between the P&ID and network diagram that lists which block goes into which device and what loops the blocks in a device belong to. To correlate the devices in the network drawing to the function in the P&I diagram it may be a good idea to print these cross-reference lists. Lists can typically be exported to any ODBC-compliant database, such as commonly used office tools, where it can be sorted and filtered (figure 8-2).



	A	B	H	I	J	K
1	BlockTag	BlockType	ExecutionTime	ControlModuleTag	DeviceTag	VFDNumber
2	LT-122	Analog Input	34	Level Measurement	LIT-122	2
3	TT-123	Analog Input	34	Temperature Control	TIT-123	2
4	TC-123	PID Control	67	Temperature Control	TIY-123	2
5	TY-123	Analog Output	40	Temperature Control	TIY-123	2
6	_R_LD	Resource Block	0		LIT-122	2
7	_T_LD	Transducer	0		LIT-122	2
8	_D_LD	Display	0		LIT-122	2
9	_R_TT	Resource Block	0		TIT-123	2
10	_T_TT	Transducer	0		TIT-123	2
11	_D_TT	Display	0		TIT-123	2
12	_R_FI	Resource Block	0		TIY-123	2
13	_TF_FI	Transducer	0		TIY-123	2
14	_D_FI	Display	0		TIY-123	2

Figure 8-2. Block execution allocation exported to Microsoft® Excel.

P&I Diagram

P&I diagrams usually follow the ANSI/ISA-5.1-1984 (R1992) standard, which states that a P&ID is used for symbolizing the key functions of an instrument whereas other details are covered elsewhere. The fieldbus technologies do not change this, that is, the separation of hardware from control functionality. Just as in the past, the P&ID shows neither wiring aspects nor which device a function block is being executed in. It is possible to indicate where control is done by placing the control bubble next to the bubble of the device where it is executed, for example, next to the valve bubble. Use consistent tagging regardless of whether the blocks are of a proprietary nature in a central controller, in PROFIBUS PA, or in FOUNDATION blocks in a field device because control is independent of the physical location. Function blocks perform the same job regardless of where they are located. Thus, there is no need to indicate which device the block is in on the P&ID; this is done in a separate document. Keeping the transducer blocks and resource (physical) blocks that are related to device configuration out of the P&I diagram ensures that the documents are not cluttered.

Signal linking between fieldbus devices as shown on P&ID's and other system documentation needs to be depicted differently from signal linking for conventional analog electronic signals. At the time of this writing, the only symbol available for use that comes close to being appropriate in ANSI/ISA 5.1 (1984) R-1992 is the data link symbol typically used for DCS applications. To the end that this symbol is not specific enough, the committee for that standard is addressing the need for a specific symbol as part of their current revision work. It is anticipated that the revised standard will be issued by the end of 2001 or early 2002. In the meantime, it is recommended that the current data link symbol be used.

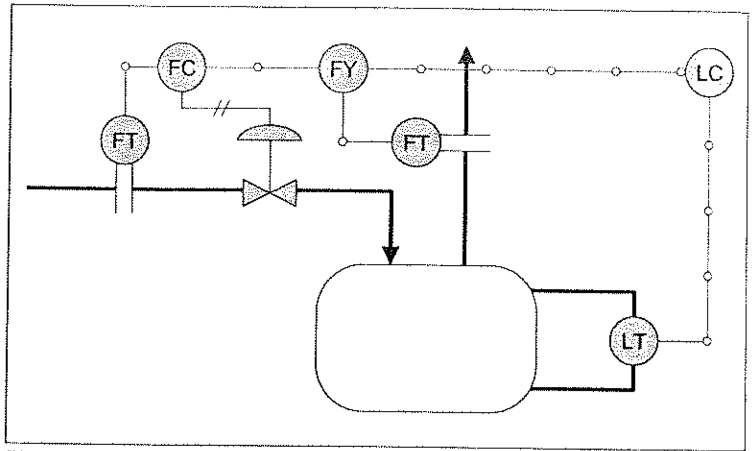


Figure 8-3. P&ID detail using data link symbol for fieldbus interconnections.

There may or may not be an exact correspondence between the "bubbles" in an ISA-style P&ID and the function blocks entered in the control strategy. A single ISA bubble for calculation in the P&ID may require that more than one arithmetic function block be configured, whereas in other cases a single function block performs the function of several symbols.

Network Drawing

Rather than use a number of traditional loop diagrams it is better to document the entire fieldbus network of maybe twelve devices or more in a single network drawing (figure 8-4). The network drawing shows electrical aspects of the network, including safety barriers, junction boxes, and communication interface port assignment, not where control functions have been allocated (figure 8-5). From a network drawing it is possible to tell which devices will be affected if the wire is disconnected. If individual loop diagrams were used technicians could be misled into believing that disconnecting would only affect the single loop.

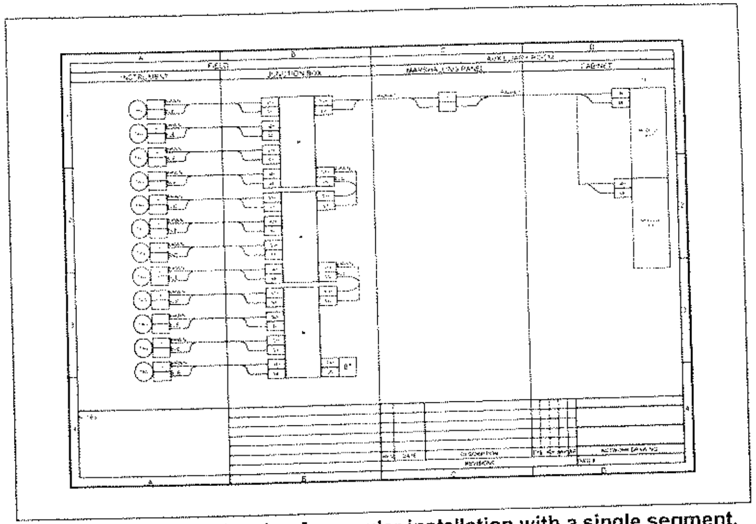


Figure 8-4. Network drawing for regular installation with a single segment.

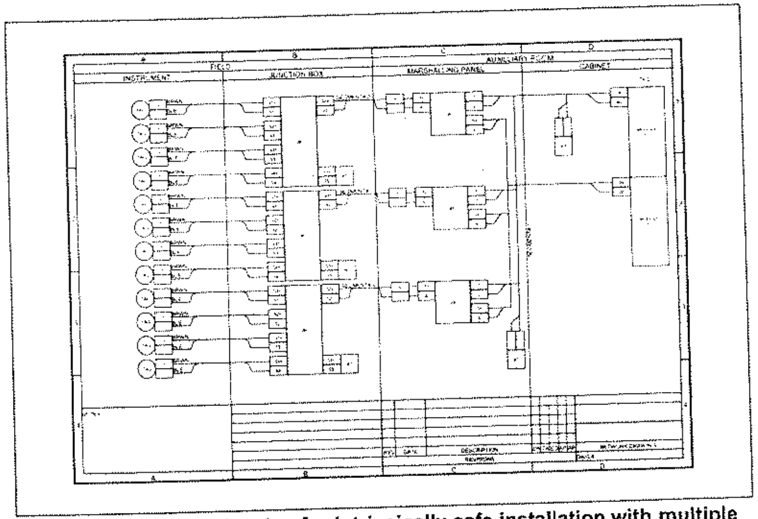


Figure 8-5. Network drawing for intrinsically safe installation with multiple segments.

For reference purposes, each fieldbus network in the plant should be numbered. Name the networks around the plant according to some logical tagging scheme, for example, according to linking device number and port number: FN-ii-pp. Use the network tag name in the network drawing as well as in the configuration tool for easy cross-referencing. Use consistent tags for fieldbus and conventional devices. Don't include network numbers in the device tag as it is not always clear which network a device will eventually be on.

Represent fieldbus and nonfieldbus devices consistently in network diagrams and conventional loop diagrams. These documents are used for troubleshooting devices, and fieldbus devices have the same basic function as conventional devices.

Device and Network Documentation

The fieldbus technologies make it possible to simplify device documentation, such as instrument summary lists, because the information is not trapped in the device or handheld. Many configuration tools today allow all information about the devices and other parts of the system to be exported to any ODBC-compliant database. This includes general information like device tag, manufacturer, device type, revision, unique ID (hardware address), type, and which network they are on. In addition, the complete resource, transducer, and function block configuration are also exported (figure 8-6).

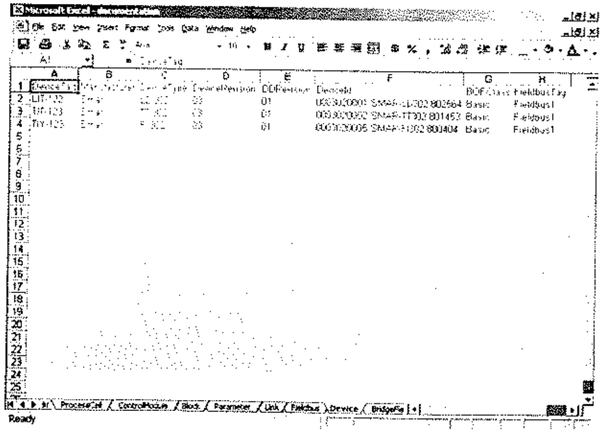


Figure 8-6. Instrument summary listing and network association exported to Microsoft® Excel.

It is a good idea to use the fieldbus configuration tool to maintain electronic documentation in a paperless system.

Control Strategy Documentation

FOUNDATION Fieldbus specifies a graphical programming language, and consequently most configuration tools are able to print out the function block diagram as part of the configuration documentation (figure 8-7).

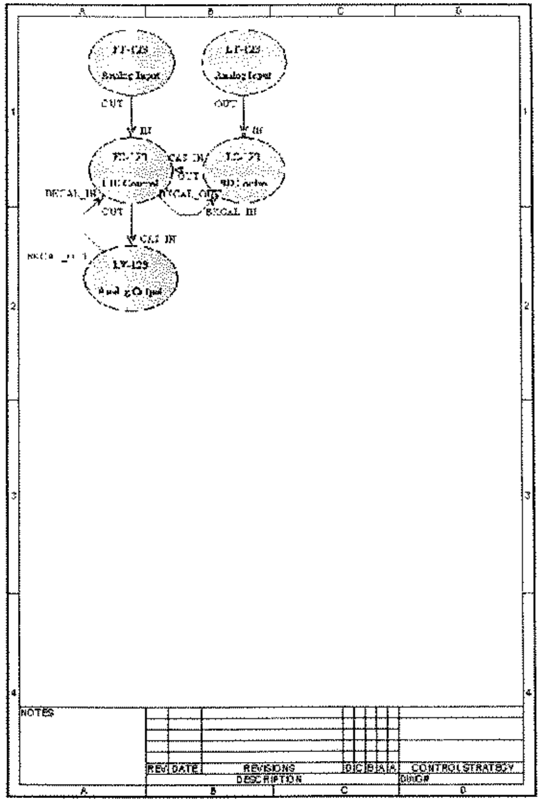


Figure 8-7. Control strategy.

Many other aspects of the strategy configuration can also be documented. For example, the engineering tool will show the macrocycle each network will have. The configuration tool automatically generates cross-reference documentation showing to which device the blocks have been allocated. The tag of the device in which a block is executed is shown next to the block itself.

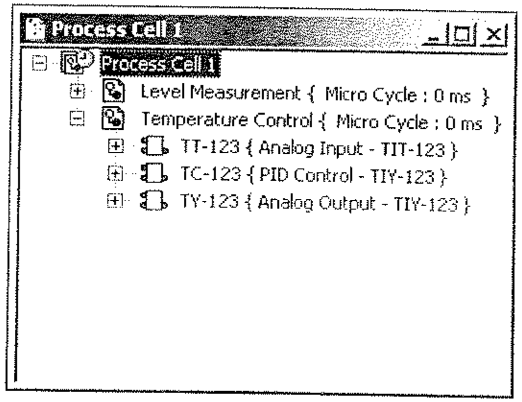


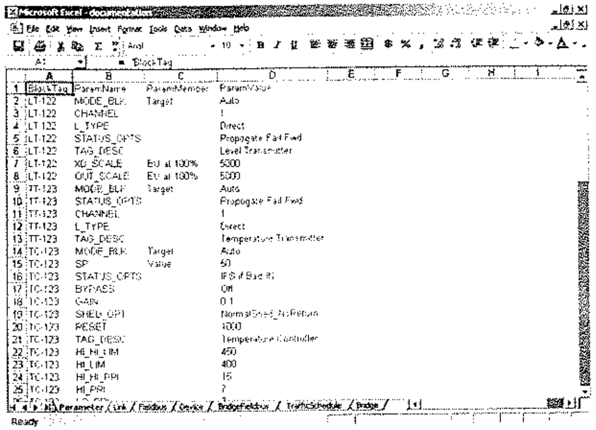
Figure 8-8. Tag based indication of device allocation for blocks in a loop.

Usually, the configuration tool has a search function that locates a function block in the control strategy or device on the network. When the control strategy is built, a database is generated from which the configuration of each device is later downloaded to the instruments in the field. The parameterization of every block can be printed and exported to other applications (figure 8-9).

Factory Acceptance Test (FAT)

Factory acceptance tests (FATs) are an essential quality assurance check to verify that all the components of the host are working properly. For FOUNDATION Fieldbus, a substantial part of the control strategy is executed in the field devices. Similarly, for PROFIBUS PA key functions for the control loop are performed in the input and output function blocks in the field devices. Therefore, it is not possible to test the control strategy without connecting all field instruments. A number of test plan approaches are possible, which are presented in the following paragraphs.

A *pure host test* would test only the system from the field-level interfaces up to the workstations, that is, without connecting the field instruments. Only a single device would be connected to the interface ports one by one to check the basic communication of each port. Subsequently, an interoperability test could be carried out to test communication against one device of every type. FATs of field devices would be done at the respective instrument manufacturer's facilities, just as in the past. A full system test would have to wait until the SAT. The installation and commissioning of instruments can be done in parallel to reduce the project time. For hardware verification, this is sufficient because there is no I/O-subsystem point mapping, scaling, or calibration to be checked.



	A	B	C	D
1	BlockTag	ParamName	ParamMember	ParamValue
2	LT-122	MODE_BLK	Target	Auto
3	LT-122	CHANNEL		1
4	LT-122	L_TYPE		Direct
5	LT-122	STATUS_OPTS		Propogate Fail Fwd
6	LT-122	TAG_DESC		Level Transmitter
7	LT-122	XD_SCALE	EU at 100%	5000
8	LT-122	OUT_SCALE	EU at 100%	5000
9	TT-123	MODE_BLK	Target	Auto
10	TT-123	STATUS_OPTS		Propogate Fail Fwd
11	TT-123	CHANNEL		1
12	TT-123	L_TYPE		Direct
13	TT-123	TAG_DESC		Temperature Transmitter
14	TC-123	MODE_BLK	Target	Auto
15	TC-123	SP	Value	60
16	TC-123	STATUS_OPTS		IFS if Bad IN
17	TC-123	BYPASS		Off
18	TC-123	GAIN		0.1
19	TC-123	SHED_OPT		NormalShed_NoReturn
20	TC-123	RESET		1000
21	TC-123	TAG_DESC		Temperature Controller
22	TC-123	HI_HI_LIM		450
23	TC-123	HI_LIM		400
24	TC-123	HI_HI_PRI		15
25	TC-123	HI_PRI		7

Figure 8-9. Block parameterization exported to Microsoft® Excel.

A *full system test* would verify the system hardware and configuration in its entirety, requiring every field instrument to be connected and the complete system configuration to be downloaded. This type of test would require all instruments to be manufactured and sent to the FAT site well before test can begin. The testing facilities would be required to rig all instruments. The installation and commissioning of instruments can only start after the FAT has been completed. For a large system, a full system FAT would have to wait for every instrument to arrive. This test offers the advantage that all issues, including the configuration of field devices, would be resolved by the time the system arrives on site.

A *network test* would verify the system one field network after another. A smaller set of field devices would be required. These would be connected to one network at a time in their actual quantities, and the configuration for the network would be downloaded. This type of test would not require as many instruments to be manufactured before the test could begin and would require fewer staging resources.

Decide at an early stage of the project which form of FAT should be performed. During the FAT, the complete host, including all linking devices, controllers, and conventional I/O subsystems, as well as the host-level network should be subjected to both hardware and software functional tests. The conventional part of the system should be tested the same way it was before. Many of the checks performed during conventional FAT also apply to the fieldbus part, for example, testing the quality of terminations, and so on.

FAT Test Plan Additions

Together, the buyer and supplier usually work out a FAT test schedule document. For the fieldbus part of the system, there is no I/O calibration and scaling to check and no point mapping to database to verify. However, there are a number of new points that must be performed:

- Ping all the host-level network nodes.
- Test all field-level network communication ports against a known good device. Simply make sure the device appears in the live list.
- Test interoperability with one field-level device of each type.
- Test switchover from the primary to secondary field-level communications interface.
- Test switchover from the primary to the secondary host-level network.

During a full system test performed either at FAT or SAT, you should check the actual control strategy against the drawings and functional specification. When the FOUNDATION fieldbus programming language has been used throughout the system the testing can focus on the regulatory aspects since the interlocks for reset windup protection and bumpless transfer are built in. You should also check that the proper configuration for selecting safety versus availability has been done. Similarly, you should verify any additional external interlock. Input values and fault conditions can be simulated in the field devices using their simulation and loop test functions. Finally, you should also confirm output shutdown in response to simulated transmitter, interface module or process fault, and linking device and field power failure.

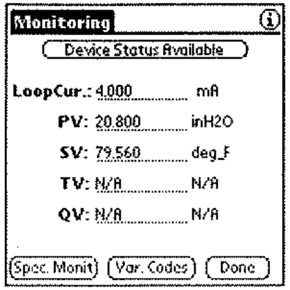
Site Acceptance Test (SAT)

Site acceptance tests (SAT) are an essential quality assurance check to verify that the correct devices are in place and that they are configured and working properly.

SAT Test Plan Additions

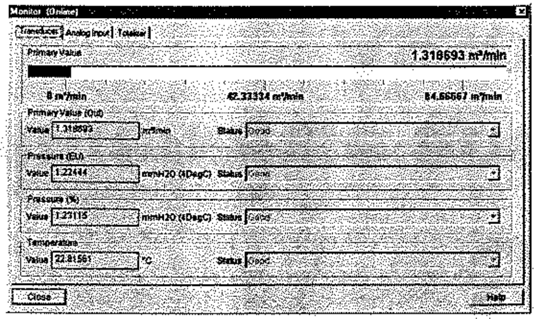
The buyer and supplier usually work out an SAT test schedule document together. If the configuration has not been downloaded and tested during the FAT it should be during the SAT. There are a number of new points to be tested in the field:

- Make sure all field devices appear in the live list for the relevant network.
- Ensure that the correct device is in the right place by verifying that the physical readings from sensors and to actuators are coming through.



LoopCur.: 4.000 mA
PV: 20.800 inH2O
SV: 79.560 deg_F
TV: N/A N/A
QV: N/A N/A

Figure 8-10. Remote monitoring of HART device.



Primary Value 1.318693 m³/min

Primary Value (Out)
Value 1.318693 m³/min Status Good

Pressure (EU)
Value 1.22444 mmH2O (4DegC) Status Good

Pressure (%)
Value 1.23115 mmH2O (4DegC) Status Good

Temperature
Value 22.81561 °C Status Good

Figure 8-11. Remote monitoring of PROFIBUS PA device.

EXERCISES

8.1 Is it necessary to make a VCR calculation at the time of engineering?

8.2 How many device support files are provided with FOUNDATION devices to make it possible for the engineering tool to configure them?

8.3 During the purchasing process, should the function block requirement be specified on a per-tag basis?

8.4 Should physical device details be indicated in the P&I diagram or in the network diagram?

8.5 Can preengineered FOUNDATION control strategy templates be used with any device?

BIBLIOGRAPHY

1. ANSI/ISA-5.1-1984 (R1992), *Instrumentation Symbols and Identification*, Research Triangle Park, NC: ISA, ISBN 0-87664-844-8, 1992.

2. ISA-20-1981, *Specification Forms for Process Measurement and Control Instruments, Primary Elements and Control Valves*, Research Triangle Park, NC: ISA, ISBN 0-87664-347-0, 1981.

9

Maintenance and Asset Management

Maintenance and asset management for fieldbus concerns the health of the instrumentation and to some extent the networking hardware, but does not involve the control strategy. However, asset management indirectly improves control because it can be used to ensure that the devices are in a better condition to perform control. Troubleshooting installation problems and debugging control strategy is covered in chapter 6, "Troubleshooting."

One of the main distinguishing features of HART, FOUNDATION Fieldbus, and PROFIBUS PA is that they have standard parameters for diagnostics, operational statistics, calibration, and identification information. This is what makes them fieldbus technologies, not just I/O networks. This feature is utilized by software that can retrieve this information without the need for custom programming, data mapping, or display. Asset management functions like calibration and diagnostics may be included as part of the configuration tool, or they may be performed by stand-alone software. To accountants, asset management means tracking the expenses related to machinery, process equipment, buildings and vehicles, and the like. These subjects are beyond the scope of this chapter.

The focus of this chapter is the network-enabled management of instrument assets. Independent of the maintenance scheme used in your plant you can use the fieldbus technologies to obtain information from the field devices that will simplify maintenance and make it more efficient. Ultimately, it is a good idea to use network-enabled asset management to switch to a proactive maintenance program.

When your plant embarks on the first project involving fieldbus technologies work with the quality assurance department. Define how the maintenance tool shall be used and how to make it an integral part of the calibration and service procedures.

Reactive maintenance is a scheme in which a device is only fixed after it is broken, or rather only after it has been found to be broken. This kind of maintenance can be very dangerous since, depending on the application of the device, the plant may not be safe while the device is not operating properly. A failed device may, again depending on its application, cause production to stop or reduce the product quality below an acceptable level. The low availability means loss of production time. Shutdowns due to failures are normally very expensive as whole batches of products often are destroyed and production capacity may be reduced for a long time. Stopping and restarting the process every time there is a failure is also frustrating. Time, raw materials, energy, and other resources are wasted. If the process goes wrong it can be very time consuming to clean up the mess. Although reactive maintenance is far from the preferred maintenance scheme, use the fieldbus technologies to improve the situation. Rely on the field instruments' ability to report faults to the host through standard means. This will make it possible for maintenance technicians to take quicker action.

Preventive maintenance is a scheme in which devices are serviced at a chosen interval whether they need it or not. This approach improves safety and availability since surprise stops are in many cases avoided. However, this scheme is also costly because resources like time and spare parts are wasted on many devices that actually need no attention, that is, unnecessary maintenance. The preventive maintenance is disruptive since production may not be able to run normally while the maintenance is carried out. Although availability is not at its optimal level use the fieldbus technologies to store and retrieve information about the last calibration or service in the field device.

Predictive maintenance is also a scheme in which service is performed on a periodic basis whether the device needs it or not. However, the service interval is optimized using failure rate and drift statistics gathered for each device type over long periods of time. There is still some waste of resources, but less of it since unnecessary maintenance is reduced. Use the fieldbus technologies to more accurately determine when devices fail and how much they drift.

Proactive maintenance is a scheme in which service is targeted toward the instruments that really need it, that is, based on the condition of the device. Resources are not wasted on devices that need no maintenance. This scheme requires a minimum of resources, has the lowest possible cost, and offers the most efficient use of manpower. Use fieldbus technologies in addition to diagnostics to continuously monitor "leading indicators" of wear and tear, stress, extreme ambient conditions, and the number of operations to predict future faults without the need for manual data collection and entry. Network-enabled asset management is faster and easier than manual data collection and entry as there is no delay between the time the diagnostic data is created and the moment when it reaches the asset management software.

Asset Management-enabled Systems

Just adding a fieldbus interface module to the system hardware does not open up the great potential of these technologies. Asset management is a software-centric approach to maintenance and calibration. To fully benefit from the fieldbus technologies make sure the engineering tool or dedicated asset management software is permanently connected to all field-level networks so there is continuous online overview and monitoring of the devices even while the plant is operating. Diagnostics, configuration, identification, and the like can be done any time. Although a stand-alone or temporarily attached configuration tool may work for a small pilot plant, it prohibits effective utilization of the tool for asset management in a larger plant. Inconsistencies between the control and device databases often may result. It may be inconvenient to have to move between different workstations. It is a good idea to leave your maintenance application running continuously on the same workstation as the process visualization software. When a "Bad" status, block error, or abnormal mode is indicated in the operation screen, staff can check the maintenance tool to see what is going on without having to leave the station (figure 9-1).

Asset management deals with devices. For that reason, it is the resource (physical) block and the transducer blocks in FOUNDATION or PROFIBUS PA devices that are accessed. The details of the parameters in these blocks are covered in chapter 4, "Configuration."

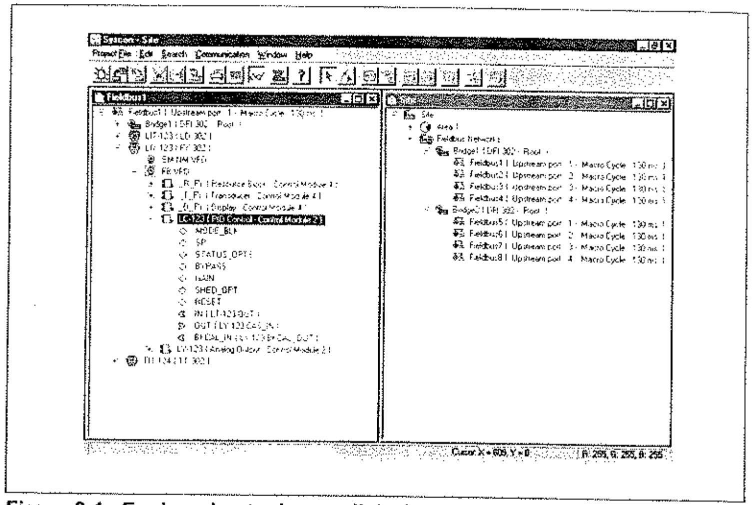


Figure 9-1. Engineering tool sees all devices in the plant.

Calibrating Fieldbus Devices

Calibration is typically done on the workbench in the instrument workshop. Some simple calibration that does not require bulky reference standard equipment can be carried out in the field. Since detailed diagnostics and calibration require communication with the device communications facilities should be available in the instrument workshop. For a large installation, you can justify a small stand-alone maintenance station consisting of a workstation and a simple interface with power supply. This is easiest done using a linking device that has integrated power (figure 3-11).

For a small installation a dedicated system for the workshop may not be justified. In this case, an alternative is to run one field-level network from the main system into the instrument shop for the sole purpose of calibration and maintenance. Similarly, the host-level network can also run into the shop to a workstation dedicated to maintenance tasks. It is a good idea to have the maintenance station connected to the host-level network so any configuration changes and calibration record entries are stored in the same single database.

Calibration versus Range Setting

The fieldbus technologies do not change the concept of calibration. Do not confuse calibration with range setting; they are two different things. For analog transmitters, calibration and range setting

was performed using the same set of potentiometers. Therefore, the terms *calibration* and *range settings* have been confused. To separate the two functions the true calibration in HART is called "trim." In FOUNDATION Fieldbus and PROFIBUS PA true calibration is indeed called "calibration," whereas range setting is called "scale." For example, transducer block calibration is not used to cancel pressure from a "wet leg" in, for example, level or flow measurement. Rather, this is done with the transducer scale in the analog input function block. Range setting can be done remotely, but calibration cannot because it by definition requires connection of standard reference input. Calibration must occur at the device in order to apply different known input values one by one to calibrate against. Since HART transmitters and positioners have 4-20 mA output and input, respectively, the loop current can also be calibrated, though this is very rarely done. Therefore, HART devices also have a "current trim."

Calibration is the correction of sensor reading and physical outputs so they match a standard. Therefore, calibration means that a known external standard input has to be applied when performing input calibration and that the output has to be measured for output calibration. In other words, calibration cannot be performed remotely because it requires someone to connect the external standard. Use calibration when the primary value reading (in engineering units) does not match the applied input. Since the primary value reading is later scaled to obtain the percentage reading (and 4-20 mA output of a HART transmitter), sensor calibration indirectly affects the percentage reading too.

Range setting for HART transmitters is done primarily to tell the transmitter the values at which the output shall be 4 mA and 20 mA. For FOUNDATION and PROFIBUS PA transmitters, range setting is typically only done for applications of inferred measurements. Range can be set remotely without applying any input or measuring the output. Use range setting when the percentage reading is not correct. Range setting does not affect the primary value reading.

Note that for HART transmitters it is possible to correct a calibration error by changing the range—and this is often the practice. The engineering unit reading is wrong, but the 4-20 mA output will still be correct.

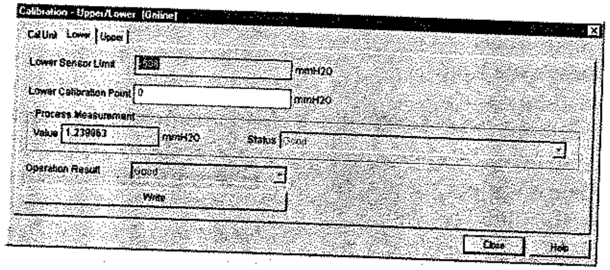
Performing Calibration

Calibration is a device function. For a FOUNDATION or PROFIBUS PA device it is therefore done from the corresponding transducer block. However, range setting is a control strategy function and is therefore done in the corresponding function block.

Although the fieldbus specifications standardize how the user interacts with the devices, the exact procedure for performing calibration may differ depending on the configuration software-handheld and the device. Refer to the relevant product manuals for details.

Input Calibration

Sometimes the primary value reading may differ from the applied input. The reason may be that, for example, the transmitter mounting position or the reading shifted because of overpressure, overtemperature, or long-term drift. Calibration is performed to match the reading with the applied input. Applying a value from a known standard and informing the transmitter what that value is "teaches" the transmitter the correct value. The correct reading is keyed into the device through the engineering tool, which usually also shows the present value as seen by the transmitter (figure 9-2). The correct value is referred to as the "calibration point," or the "trim point" in the case of HART. Calibration for FOUNDATION Fieldbus and PROFIBUS PA transmitters is performed by writing the applied input to the low calibration point (CAL_POINT_LO) and high calibration point (CAL_POINT_HI) in the transducer block. Input calibration changes the process value that may cause process upsets. Therefore, it is prudent to either calibrate the device off line or with the control loop in manual. Many configuration tools prompt the user to check mode.


Calibration - Upper/Lower (Online)

Lower Sensor Limit [illegible] mmH2O

Lower Calibration Point 0 mmH2O

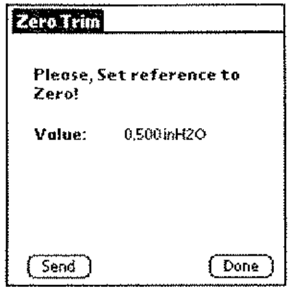
Process Measurement

Value 1.239963 mmH2O Status Good

Operation Result Good

Figure 9-2. Calibration of a PROFIBUS PA transmitter.

A special case of sensor calibration is zeroing the transmitter, which is commonly done for some measurements. A zero input is achieved to simulate the input without having to connect any standard. This is done simply by venting the pressure, shutting off the flow, or emptying the tank, as appropriate for the measurement being done. The calibration can then also be performed without having to key in any calibration point value (figure 9-3).


Zero Trim

Please, Set reference to Zero!

Value: 0.500inH2O

Figure 9-3. Zero calibration of HART transmitter.

Typically, calibration is done in two points as close as possible to the expected measurement range. These two points are referred to as the low and high calibration points. For most modern transmitters, the calibration of these two points is noninteractive. This means that one can be calibrated without affecting the other, so an iterative process is not necessary. All transmitters, whether they state it or not, have a finite measurement resolution, and a measurement over a small span would therefore result in poor accuracy in percentage terms. For most transmitters, there must therefore be a minimum span between the upper and lower calibration points. Attempts to calibrate with smaller spans are rejected. For FOUNDATION Fieldbus and PROFIBUS PA devices, the minimum span (CAL_MIN_SPAN) can be reviewed from the transducer block.

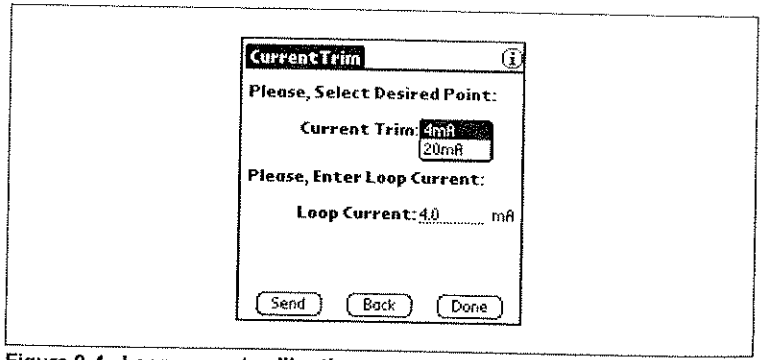
Output Calibration

Output calibration mainly applies to signal converters that have analog output such as 20-100 kPa (3-15 psi) or 4-20 mA. To calibrate the output signal, you first force the transducer block to output a signal of a set level. The actual physical output may deviate from the desired level that has been set. If it doesn't, calibration is not required. The actual physical output value as measured externally is then written back into the device from the software-handheld, which will then correct its output accordingly. Typically, the low and high calibration points are noninteractive, which eliminates the need for an iterative process.

Control valve positioners that are able to calibrate themselves are a special case of output calibration. With a simple click from the engineering tool, the positioner will stroke the valve over its entire range of travel and automatically calibrate itself for the open and closed end points. This process may take a minute or a few, and once calibrated the user is informed. Obviously, stroking the valve upsets any process fluid flowing through it, and therefore calibration should only be done off line or while the process is not operating. The PROFIBUS PA actuator transducer block has a standard parameter to invoke this function, SELF_CALIB_CMD.

Current Loop Calibration (Current Trim)

HART transmitters have a 4-20 mA output. Because there are no mechanical parts not much drift is associated with the 4-20 mA. Therefore, there is no real need for current loop calibration. Nevertheless, it is possible to calibrate it in two points, which are called current zero trim and current span trim for the 4 mA and 20 mA points, respectively. When the output calibration is invoked the software-handheld forces the current output to the relevant value and the user is prompted to measure and key in the actual output (figure 9-4).



Please, Select Desired Point:

Current Trim: 4mA / 20mA

Please, Enter Loop Current:

Loop Current: 4.0 mA

Figure 9-4. Loop current calibration.

Calibration Record-keeping

A field device also stores the information that is pertinent to the calibration. From the engineering tool, you can review the calibration record to determine the calibration status of the device. For example, HART has a parameter for a date that can be used to record the time of the last calibration or the time scheduled for the next one, based on the manufacturer's recommendation or plant experience. PROFIBUS PA devices have an installation date parameter in the physical block that you can perhaps be used to store the date of the last or next calibration instead. In addition to date and time, the FOUNDATION Fieldbus transducer blocks also store who did it, where it was done, the method used, and the like. Both the FOUNDATION and PROFIBUS PA transducer blocks have the calibration point parameters. From these, you can review the last calibration values to determine if the calibration performed is suitable for the application. Because the information is stored in the device itself it stays with the device at all times and is not lost.

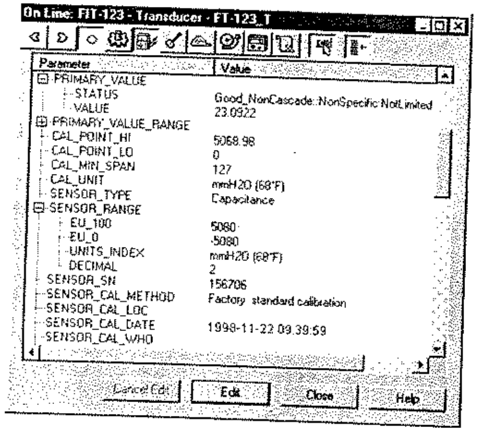
Use the calibration record capability in the calibration management scheme. Use the networking to get an overview of the calibration status of the devices around the plant. Target the resources toward the devices for which calibration is due. It is a good idea to update the calibration record information each time a calibration has been performed. Include device record-keeping as part of the plant procedures.

Information Pertinent to Calibration

Calibration can only be performed at points within the limits for the sensor, and the two points of calibration must be separated by at least the minimum span limit. These limits are stored in HART, FOUNDATION Fieldbus, and PROFIBUS PA devices along with other pertinent information, such as sensor type (figure 9-5). Use this information to assist you in the calibration.

Diagnosing Fieldbus Devices

Network-enabled diagnostics can help you troubleshoot when you are maintaining devices that have already failed right from the engineering tool. HART, FOUNDATION Fieldbus, and PROFIBUS PA devices have several parameters that are dedicated to diagnostics that simplify troubleshooting, as explained in chapter 6, "Troubleshooting." Use the diagnostics capability to spot device failures faster and therefore rectify them earlier. In other words, early detection enables you to minimize breakdown times, thus avoiding unnecessary process downtime. For example, it is a good idea to configure the process visualization software to display the parameter status and notify the operator when faults in the field devices are detected, as explained in chapter 7, "Operation."



On Line: FIT-123 - Transducer - FT-123_T

Parameter	Value
PRIMARY_VALUE	
STATUS	Good_NonCascade::NonSpecific:NotLimited
VALUE	23.0922
PRIMARY_VALUE_RANGE	
CAL_POINT_HI	5068.98
CAL_POINT_LO	0
CAL_MIN_SPAN	127
CAL_UNIT	mmH20 (68°F)
SENSOR_TYPE	Capacitance
SENSOR_RANGE	
EU_100	5080
EU_0	-5080
UNITS_INDEX	mmH20 (68°F)
DECIMAL	2
SENSOR_SN	156706
SENSOR_CAL_METHOD	Factory standard calibration
SENSOR_CAL_LOC	
SENSOR_CAL_DATE	1998-11-22 09:39:59
SENSOR_CAL_WHO	

Figure 9-5. Calibration related information.

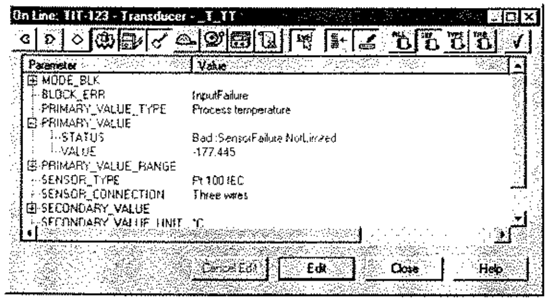
Devices based on fieldbus technologies have wide diagnostic coverage and use additional sensors, other electronics, and self-diagnostics firmware algorithms that can detect faults based on manufacturers' experience with failure modes. The transmitter detects the failure of external sensors such as a thermocouple or even faulty or poor wiring. Ambient conditions like temperature that exceeds operating limits may also be reported. When the operator is notified of the fault, he or she can immediately attend to it, which will ensure good transmitter performance. Use asset management features to do as much of the maintenance tasks as possible from the control room. Doing so will lower your exposure to dangerous plant environment and minimize the number of work permits you will require.

Reducing Unnecessary Maintenance

Perhaps the biggest advantage of continuous device diagnostics is not that it tells you a device has failed but that it tells you that it has not failed. In other words, the diagnostics provided by the fieldbus devices can help you to make an assessment confirming that no troubleshooting is required. By ruling out the devices that are functioning, you can target troubleshooting to other devices or to the process. Review the results of the self-diagnostics to determine if you have a device problem or a process problem. This will reduce the number of times a device from the field is taken unnecessarily to the shop, only to be tested and found to be OK.

Reducing Repair Time

Make it a point to always use the engineering tool to pinpoint the instrument that has failed and to determine in greater detail what the problem is before you go into the field (figure 9-6).



On Line: TIT-123 - Transducer - _T_TT

Parameter	Value
MODE_BLK	
BLOCK_ERR	InputFailure
PRIMARY_VALUE_TYPE	Process temperature
PRIMARY_VALUE	
STATUS	Bad::SensorFailure:NotLimited
VALUE	-177.445
PRIMARY_VALUE_RANGE	
SENSOR_TYPE	Pt 100 IEC
SENSOR_CONNECTION	Three wires
SECONDARY_VALUE	
SECONDARY_VALUE_UNIT	°C

Figure 9-6. Diagnostics.

Armed with this information, you can carry the right spares and tools into the field and perhaps fix the problem on the spot rather than bring the instrument back to the workshop. For FOUNDATION Fieldbus, the block error (BLOCK_ERR) parameter is ideal for gaining a diagnostic overview and is used by asset management software to check device health.

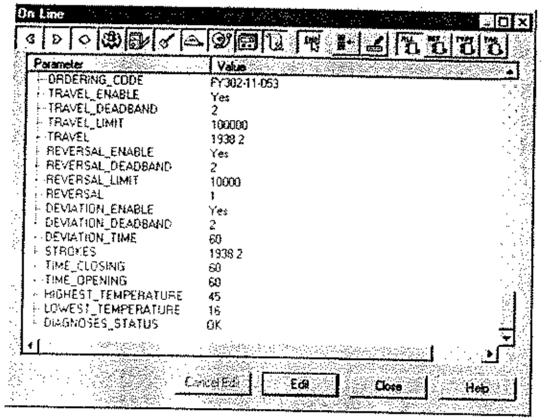
Predicting Failures in Fieldbus Devices

To a much larger extent than diagnostics can, continuous operational statistics enable plants to move to a proactive maintenance scheme. Diagnostics tell you if a device has already failed, but leading indicators collected in the field instruments like wear-and-tear data and exceeded operating conditions can be used to trip maintenance alarms before the device fails. Statistics predict degrading device performance, which can cause inaccuracies and faults. Plants can therefore use operational statistics to target maintenance resources on devices that may soon be in need of repair.

Operational Statistics

Operational statistics are stored in the transducer blocks. Control valve positioners are a good example of a device that collects a large amount of operational statistics. Examples of these are total valve travel (odometer) and the number of reversals. Plants can use such information to predict the time of failure by comparing these statistics to the life expectancy data for critical parts as provided by the manufacturer. Plants can use operational statistics to more accurately predict when maintenance is due since wear and tear is measured rather than estimated on an average. For example, the PROFIBUS PA valve transducer has standard parameters for total travel in the equivalent number of full strokes and a total travel limit at which maintenance should be performed.

Devices that trigger an alarm when a set limit for the operational statistics has been exceeded provide early warning. As part of your new maintenance procedures, make sure you use these tools to tap into the information of the instruments. This will give you instant access to the status of any device at any time and thus a complete picture of the entire plant. Obtain information from the equipment manufacturers on the recommended number of cycles before replacement is needed and configure that number into the device's wear-and-tear alarm (figure 9-7).



Parameter	Value
ORDERING_CODE	FY302-11-053
TRAVEL_ENABLE	Yes
TRAVEL_DEADBAND	2
TRAVEL_LIMIT	100000
TRAVEL	1938 2
REVERSAL_ENABLE	Yes
REVERSAL_DEADBAND	2
REVERSAL_LIMIT	10000
REVERSAL	1
DEVIATION_ENABLE	Yes
DEVIATION_DEADBAND	2
DEVIATION_TIME	60
STROKES	1938 2
TIME_CLOSING	60
TIME_OPENING	60
HIGHEST_TEMPERATURE	45
LOWEST_TEMPERATURE	16
DIAGNOSES_STATUS	OK

Figure 9-7. Operational statistics are easily reviewed and trigger alarms notifying valve is due for service.

Make sure you configure the host to enunciate maintenance alarms to the operators. Using the engineering tool, it is easy to determine when several devices may be relatively close to needing service. Include as part of the maintenance procedures steps for keeping a record of the number of reversals performed between service and to reset the statistics counters after repair.

For example, if the manufacturer recommends that you replace the valve stem packing after a certain number of reversals, key this number into the positioner as an alarm limit. When the limit is exceeded, the alarm alerts the technicians that replacement is due (figure 9-8). In this way, you can drastically reduce the number of unscheduled shutdowns caused by unforeseen failures. Knowing about imminent failure in advance helps you reduce the chances of a surprise shutdown, and spares can be ordered beforehand. Maintenance can be proactively scheduled at a convenient time, when all the devices on a unit can be serviced together.

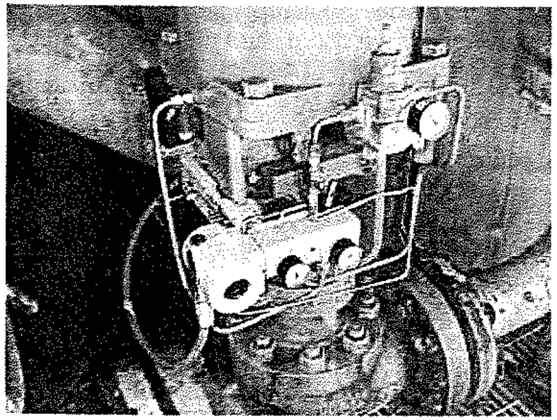


Figure 9-8. A fieldbus valve positioner diagnoses the valve and keeps operational statistics. (Courtesy of Smar)

Spot Process Variability

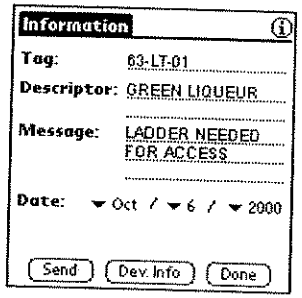
Another major benefit of operational statistics is that it gives you the ability to spot poorly tuned control loops. A rapid increase in the number of reversals is a clear indication that the loop is oscillating. Operational statistics can thus help you not only in maintenance but also in optimizing the process by indicating that it's time for tuning or a different control strategy. It is therefore a good idea to review the operational statistics from time to time in order to spot unstable behavior and thus reduce process variability.

Information from Fieldbus Devices

Fieldbus devices store data pertinent to themselves and their application. This information does not affect the operation of the device but is useful for improving maintenance practices.

Identification

HART, FOUNDATION Fieldbus, and PROFIBUS PA all have standard parameters for device tag, manufacturer, descriptor, device type, revision, serial numbers and unique ID (hardware address), and so on. Use the configuration tool at the time the device is commissioned as well as during normal operation to identify the device. HART devices accept up to eight characters for tag, sixteen characters for a descriptor, and thirty-two characters for a message (figure 9-9).



Tag: 63-LT-01

Descriptor: GREEN LIQUEUR

Message: LADDER NEEDED FOR ACCESS

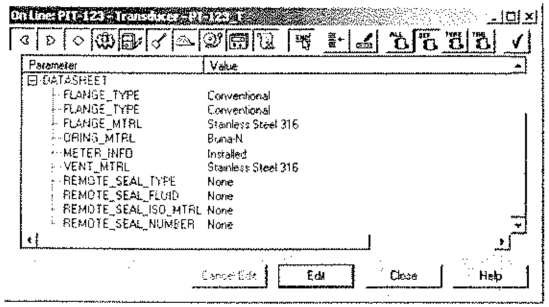
Date: Oct / 6 / 2000

Figure 9-9. Basic information from a HART device.

It is good practice to always enter the device tag, descriptor, and message and to update it whenever changes are made.

Materials of Construction

From fieldbus devices it is possible to retrieve other useful information like materials of construction for the parts wetted by the process in a pressure transmitter. You can then use this information to order identical devices or to assess its suitability for a certain application. Standard parameters in FOUNDATION Fieldbus and PROFIBUS PA include the material for the sensor diaphragm and the fill fluid. Specific parameters are used in the HART protocol as well as for other parts (figure 9-10).



On Line: PIT-123 - Transducer - PT-123_T

DATASHEET

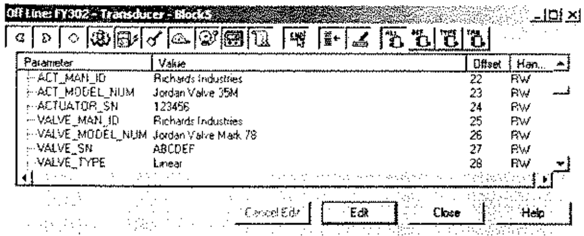
Parameter	Value
FLANGE_TYPE	Conventional
FLANGE_TYPE	Conventional
FLANGE_MTRL	Stainless Steel 316
ORING_MTRL	Buna-N
METER_INFO	Installed
VENT_MTRL	Stainless Steel 316
REMOTE_SEAL_TYPE	None
REMOTE_SEAL_FLUID	None
REMOTE_SEAL_ISO_MTRL	None
REMOTE_SEAL_NUMBER	None

Figure 9-10. Pressure transmitter materials of construction.

It is good practice to update these parameters with any changes that have been made in the actual parts to keep the record current.

Valve and Actuator Information

Fieldbus valve positioners have standard parameters for storing information not only about itself but also about the rest of the valve package it is part of. The FOUNDATION transducer block contains standard parameters for the manufacturer, model, and serial number of the valve and actuator (figure 9-11).



Off Line: FY302 - Transducer - Block5

Parameter	Value	Offset	Han...
ACT_MAN_ID	Richards Industries	22	RW
ACT_MODEL_NUM	Jordan Valve 35M	23	RW
ACTUATOR_SN	123456	24	RW
VALVE_MAN_ID	Richards Industries	25	RW
VALVE_MODEL_NUM	Jordan Valve Mark 78	26	RW
VALVE_SN	ABCDEF	27	RW
VALVE_TYPE	Linear	28	RW

Figure 9-11. Valve and actuator information for a FOUNDATION positioner.

Replacing Fieldbus Devices

Field instruments do fail from time to time and will eventually need to be replaced. Once you have connected a new device you must download the configuration to it. In this situation, you can download the configuration just for the replaced device without having to download it into all the devices on the network. Since the device's entire configuration is downloaded in one go it is quickly brought on line.

Interoperability versus Interchangeability

When you replace a failed device with an identical model you can simply download the configuration. However, if a different device type or even another version takes the place of the failed device you will have to consider the portability of the configuration. For HART devices, most of the essential parameters governing the operation are covered by universal and common practice commands that will work with any device without interoperability problems.

For FOUNDATION Fieldbus and PROFIBUS PA, basically every kind of device has a specific extension to the standard set of parameters in the transducer blocks. If a new type of device replaces an existing one you will have to create a new transducer block in the configuration but it is easy as only very few parameters have to be set, often only the mode. The resource (physical) block usually has no extensions and is supported by all devices. Thus, it will most likely not cause any interoperability problems. However, many devices have enhanced function blocks that have their own specific parameters in addition to the standard parameters. A device may also have completely proprietary blocks for special functions. If you have to replace a device in which a manufacturer-specific block is being used with a device that does not have such a block, you can allocate the function block to another device.

Alternatively, you can change the control strategy to perform the same function by using an equivalent block or set of blocks. The same applies for blocks that have proprietary parameters, though this involves additional work. It is therefore a good idea to use standard blocks from the protocol because other devices often support the exact same blocks. Likewise, different devices support different types and quantities of the function blocks that make up the control strategy. Some FOUNDATION Fieldbus device types have a limited set of function blocks in their library and a fixed quantity of each that is supported. It will be difficult for a device with a fixed block set to replace another device because chances are the two devices do not support the same blocks. However, a device that supports dynamically instantiable blocks has more flexibility to replace other devices because these blocks usually have a larger library of function blocks. As a result, there is a greater chance that the new device supports the blocks used in the old device. This is particularly true if the standard FOUNDATION blocks are used.

EXERCISES

9.1 Is calibration performed in the function blocks or in the transducer blocks?

9.2 Which parameter in the FOUNDATION function block contains diagnostics?

9.3 How can operational statistics be used to spot process variability?

9.4 Are blocks interchangeable?

BIBLIOGRAPHY

1. ANSI/ISA-51.1-1979 (R1993), *Process Instrumentation Technology*, Research Triangle Park, NC: ISA, ISBN 0-87664-390-4, 1993.

10

Availability and Safety

There is a large gap between the high-level safety provided by a safety-related shutdown system and that provided by a traditional basic control system like a DCS. Similarly, there is a great gap between the sophisticated control provided by a basic control system and the simple functions of a shutdown system. Plants can use the fieldbus technologies to fill this void by letting basic process control systems provide exceptional control strategies, great availability, and increased safety throughout the system life cycle. Using fieldbus technology, plants can also employ additional measures and installation practices to increase the availability and safety of a system. Intrinsic safety is covered in chapter 3, "Installation and Commissioning."

Fault-tolerant versus Safe Systems

Availability and safety are not the same thing. In fact, they are usually two conflicting goals. The goal of availability is to keep the process running, whereas the goal of safety is to shut the process down. The process control system (PCS) performs control that is traditionally targeted toward high availability, which is distinct from the safety-related system that is targeted toward a high level of safety.

Goals

People often mistakenly use the term *safety* when they are really referring to availability. A system without redundancy can be completely safe. For example, if the network fails it is still safe as long as the affected loops are shut down, stopping the process. Redundancy is a measure for increasing availability, not safety. For more information, see the book *Control System Safety Evaluation and Reliability,* by W. M. Goble (2d ed., ISA, 1998, ISBN 1-55617-636-8).

Safety Goals

To minimize the risk of harming people, the environment, property and the like, some control loops must be shut down if the process experiences a problem or if the system fails. In other words, there are two safety aspects: process problems and instrument failures. A process-problem shutdown is part of the interlocks in the process control system or in a dedicated safety-related system. Some loops are more risky or hazardous than others, necessitating greater safety. Safety-related systems are specifically designed to shut down dangerous loops in the event such a problem is detected. Safety-related logic solvers and instruments are usually certified by independent test institutes who assign two different levels of safety, Requirement Class (RC or AK) or Safety Integrity Level (SIL).

WARNING—Those sites that have process units that have been assessed to have risk or hazard should provide safety protection by using an approved safety-related system of the appropriate requirement class in addition to the basic process control system used for the loops associated with the danger.

In order for process problems to be detected reliably the instrumentation and controls must function properly. Microprocessor-based shutdown systems typically have built-in diagnostics or discrepancy-checking capability to detect their own failures and shut down in case they are unable to function safely. Fieldbus technologies are ideal for dealing with instrument failures, but presently they can only be used in basic control systems because no safety-approved devices are yet available. Independently safety-approved transmitters with HART exist, but they rely on the 4-20 mA signal. PROFIsafe technology has received a favorable report, but no approved PROFIBUS PA devices that use it are in the market at the time of writing. Indeed, in the interim most safety-related systems are using regular field instruments that lack safety approval.

There are no standards that clearly state the SIL for a particular process loop. Different loops may require SIL 3, 2, 1, or no SIL at all. For control loops that need no SIL-rated shutdown interlocks sites can use the fieldbus technologies to provide safety beyond that found in basic control systems using conventional technologies.

An important feature of safe networks is that they do not require redundancy. For example, the PROFISafe technology is targeted to SIL 3 without redundancy, and so are several nonredundant proprietary networks. For safety, the devices have a wide diagnostic coverage on the communications, which detects any error and if it finds it brings the process to a safe state.

For example, using either FOUNDATION Fieldbus or to some extent PROFIBUS PA the loop will be shut down if the fieldbus cable breaks, is short-circuited, or if there is another fault that causes communication or the power supply to fail. It is therefore safe to multidrop several devices on a network even without redundancy. To enhance safety in case of failure, sites should use detailed device diagnostics for other parts of the systems such as sensors and actuators for shutdown.

Availability Goals

To minimize production losses some control loops must be able to function even in the presence of a fault. Availability is a percentage measure of the time during which the loop is operating OK. In other words, fault tolerance is required in order to achieve good availability by minimizing downtime. Decentralization is one popular measure of fault tolerance, and redundancy is another.

System

For most loops, only a process control system is required. Some hazardous loops may require a safety-related system for shutdown.

Process Control Systems

Process control systems are designed to handle sophisticated control strategies for achieving optimal control. Safety-related systems do not support FOUNDATION Fieldbus, PROFIBUS PA, or even HART. They are therefore unable to benefit from the improved controls and asset management for condition-based maintenance that the fieldbus technologies provide. Thus, as much of the control and monitoring functionality as possible should be handled by the process control system.

Process Safety-related Systems

If based on the hazard analysis and risk classification it is found that some of the loops in the plant's SIL-rated system must be shut down, the plant should put a safety-related system in parallel with the process control system. The safety system has its own transmitters and shutoff valve, which are separate from the control system and are based on conventional technology (figure 10-1). The safety-related system monitors the dangerous loops and shuts down if something in the process goes wrong.

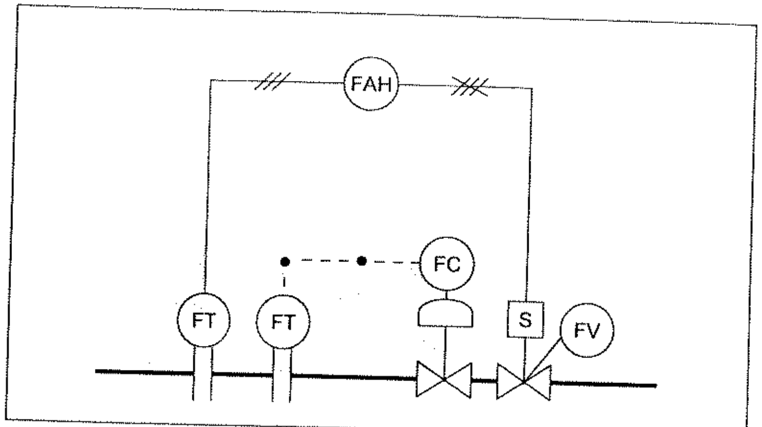


Figure 10-1. Process controls separate from safety shutdown.

Microprocessor-based safety-related systems have extensive diagnostics to make them shut down in case an internal failure is detected. However, if analog signals are used to interface with field instruments such safety-related systems are unable to access the internal diagnostics in field instruments, and therefore they miss out on the additional safety. Performing discrepancy checking through central voting catches only gross failures. Analog devices require frequent overrides so manual inspection can be performed, but that may also introduce errors.

NOTE: Non-SIL-rated field instruments can reduce the overall safety of the system to SIL 1 or lower, even though the system uses an SIL 3-rated safety-related logic solver.

Improving Availability

A high degree of availability means having a high tolerance for faults and reducing shutdowns caused by false faults. High availability is achieved through decentralization, diagnostics, and redundancy.

Availability through Decentralization

When many functions are centralized into a single piece of equipment the impact of a failure is great. For example, if thirty loops were executed in a single controller a failure would mean the loss of a significant part of the plant, possibly even a complete stop. By decentralizing functions into isolated parts a single fault affects a much smaller part of the plant. This allows the rest of the plant to continue operating, thus reducing the impact on overall availability. In other words, distribution leads to the isolation of faults, which restricts their impact. It is therefore a good idea to adopt a general philosophy of having several "small" of anything rather than a single "big." For example, the entire plant should not rely on just one network.

Network Decentralization

It is a good idea to partition large plants into areas, such that each one has a separate subnet for the host-level network. Should a fault occur, you would need to switch only the relevant subnet to a safe state. Subnets are tied together using routers. Use hub-based star topology 10/100Base-T Ethernet with only one device connected per network segment cable rather than the multidropped coax. Do not let more than thirty or so control loops depend on any central controller or linking device, even if redundancy has been implemented.

Further, subdivide each plant area into process cells and control modules such that each field-level network covers only one or a few control modules. When a failure occurs only the few loops on the field-level network would be affected. Keep the number of control loops that are dependent on a field-level network relatively low. Generally, a field-level network can only have sixteen devices, which translates into a maximum of eight loops, but it may be prudent to keep the number of control loops even lower. As a rule of thumb, for networks use eight devices to make four control loops, plus four devices for monitoring, and leave another four devices for future expansion. In other words, the field instruments in the plant are distributed over several separated field-level networks. Should a fault occur, only output devices on the respective network need to switch to a safe position. This increases system availability since a fault in one H1 network only affects a few devices.

It is also a good idea to, wherever possible, put only a single control loop that requires high availability on each field-level network. The other devices on the network should be used for less critical loops and monitoring. In this way, only a single critical loop is affected in the event of failure.

To further increase the fault tolerance of some of the field-level networks you can optionally partition a network into several segments of, say, only four devices each. However, this is rarely done for regular installations. This practice is common for intrinsically safe installations because isolating safety barriers that have built-in repeaters are generally used (figure 10-2). In such a configuration, a short circuit in a hazardous area would only render a single loop useless. Since the repeater separates the segments from each other, a fault on one segment does not affect another.

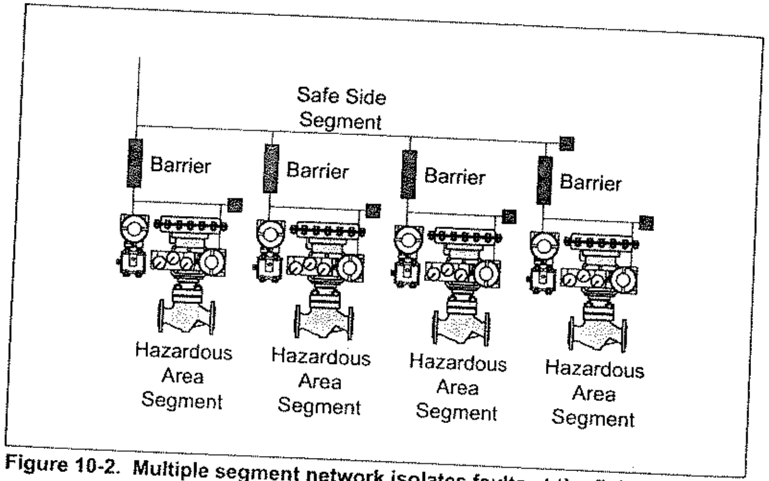


Figure 10-2. Multiple segment network isolates faults at the field level.

Additionally, in some cases it may make sense to only use single pair cables instead of multi-pair cables so damage to one cable only affects a single network.

Control Decentralization

Rather than concentrate the control of several loops into a shared centralized controller it is a good idea to use the capability of FOUNDATION Fieldbus to distribute the controls into the field devices associated with the loop. For example, a typical loop is made up of just a transmitter and a valve positioner with the latter executing the control. Use decentralized controls to increase the availability because the failure of the controlling device, for example, a valve positioner, affects only its own single loop, not thirty or more as used to be the case for loops that shared a centralized controller. This concept is called "single-loop integrity" and means that the failure of one loop does not affect another.

Confined Shutdown

You can also achieve better availability by distributing shutdown functions into the output devices in the field. Use the capability of FOUNDATION Fieldbus to allocate these functions into the field. When the device diagnostics detect a fault it will only shut down the few loops associated with the instrument. Typically only a single loop, nonrelated loops will be unaffected. For example, if a valve positioner on one network is not receiving the process variable from a transmitter on the same or other network, the loop will fail safe but not the entire network. In other words, this failure of one loop does not cause failure in other non-related loops, which will continue operating. Only degraded loops are shut down. Therefore, it is a good idea to use the FOUNDATION Fieldbus programming language.

Availability through Redundancy

The traditional way to achieve fault tolerance is to duplicate components such that a primary failure makes the secondary take over, thereby making the system more robust. Use hot-standby redundancy for the linking device (interface), power supplies, host-level network, and workstations in applications that require high availability.

Hot-standby Host-level Network Redundancy

The operator's ability to see the entire plant depends on the host-level network being up and running. Without a functional host-level network, there would be no visibility at all for the operators. Some control loops using devices on different field-level networks may also bridge across the host-level network. High availability of the host-level network is therefore very important. Any host-level network based on Ethernet can be made more fault tolerant by using simple hot-standby media redundancy adapters, which provide two paths on a single network. Mission-critical devices have two independent Ethernet ports, making possible complete redundancy on two separate networks. It is also advisable to use industrial-grade network hubs with redundant power supply.

Simple Media Redundancy

At the host-level, a plant can use simple media redundancy with any Ethernet device independently of protocol, even though the device only has a single port. No special network or redundancy management is required within the Ethernet device itself. It is therefore a good idea to use simple media redundancy for devices that only have a single Ethernet port.

Media redundancy is implemented using "splitters" that connect to a normal Ethernet port on any device, including computers, thereby providing two ports. The splitters at each device are then connected in a ring topology such that one port connects to a clockwise segment and the other to a counterclockwise segment. This forms a network with redundant paths (see figure 3-73). The ring topology is fault tolerant. If communication cannot travel in one direction it travels in the other instead. The switchover is automatic and bumpless and completely transparent to the Ethernet devices.

A similar solution uses hubs that are connected in a ring topology. Several regular Ethernet devices then connected to these hubs, which appear as normal hubs. Route the two cables for the loop through separated paths to minimize the likelihood of both being damaged or exposed to the same stress. Do not include any nonredundant segments between redundant segments.

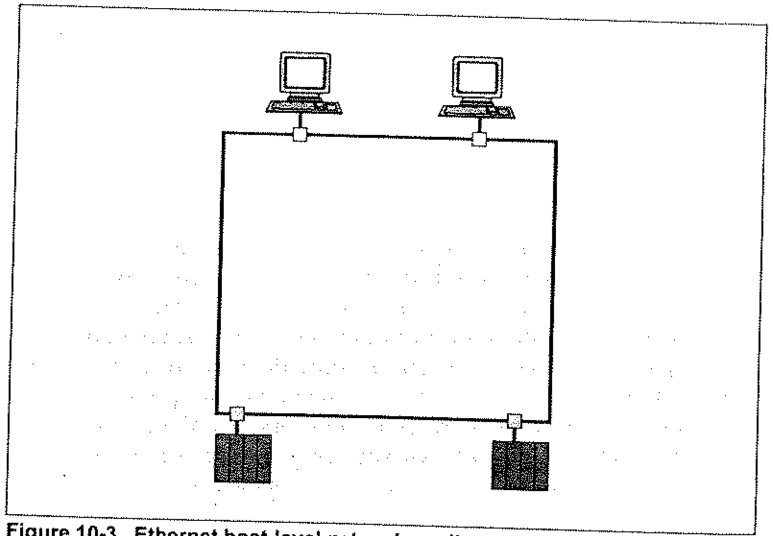


Figure 10-3. Ethernet host-level network media redundancy.

Complete Network Redundancy

The FOUNDATION Fieldbus HSE host-level protocol goes beyond media redundancy by making it possible for plants to have two complete networks in a hot-standby redundancy configuration. That is, the entire network is duplicated, including wires, hubs, and communication ports. Use full network redundancy to ensure that the system can continue operating without any loss of data, even if one device port fails (figure 10-4).

Complete network redundancy means you will need two communication ports. This can be implemented in two ways: a redundant device set may have a single port on each of the primary and secondary devices of the set, or, alternatively, the primary and secondary devices may both have two ports. In case a single device has two ports, they are labeled "A" and "B." Workstations can be fitted with dual network interface cards. Either of the two ports is used for the normal operational communication at any one time, whereas self-diagnostics is done on both ports. Depending on the health of the various parts of the network a device may use port A to communicate with some devices and port B with others.

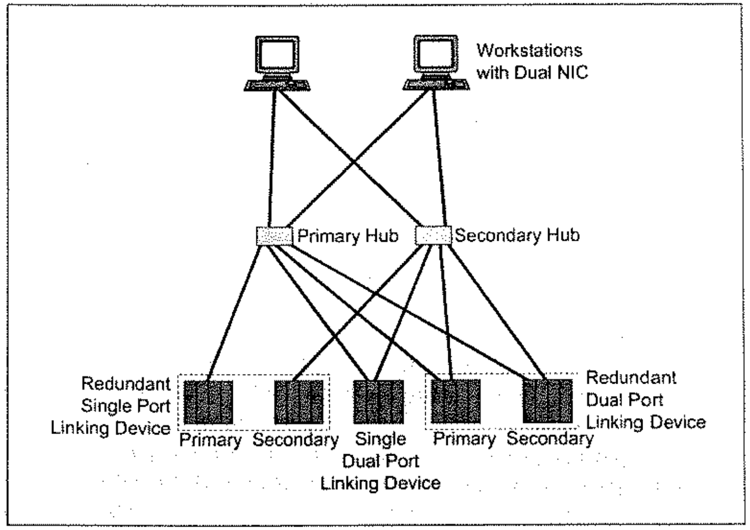


Figure 10-4. Complete Ethernet host-level network redundancy.

If a segment of the primary network is damaged, the secondary port and segment will be used instead. The diagnostics in the devices will detect port and segment failures and use the alternate path to the affected device. As a result, the system can sustain multiple faults to the devices and networks and still continue. All redundant Ethernet device pairs and the workstations are connected to both Ethernet buses. The switchover is totally bumpless and transparent to the operator, which only sees the active device in the redundant pair. Every communication port has its own unique IP address, making it possible to diagnose both the primary and secondary of a device pair and communication port. HSE devices pass standard diagnostics messages between each other on both networks. This allows every device to get a complete picture of the network's health and integrity so it can determine which port to address on any device and which path to use to get the message across.

Use complete network redundancy when high availability is required. Keep the primary and secondary of redundant device pairs on separate networks.

Network and Media Redundancy

Network redundancy can be combined with media redundancy in a dual-ring topology to effectively create four paths across the network (figure 10-5). This means that the network can sustain multiple faults but still continue to function. Thus, the networking is extremely reliable.

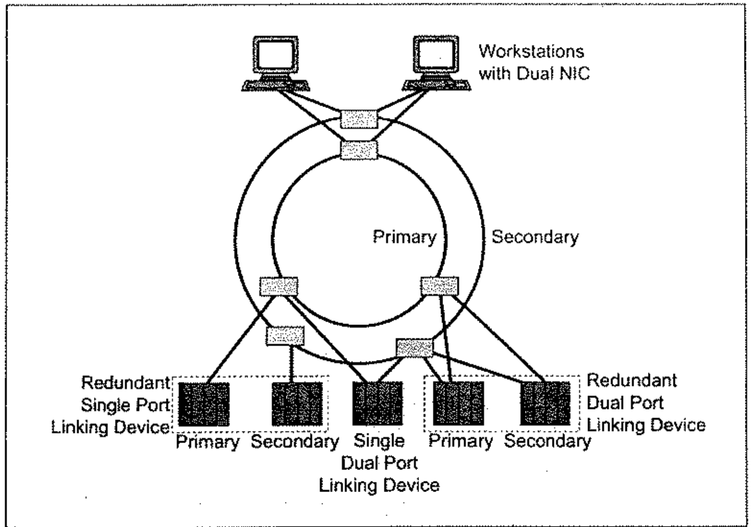


Figure 10-5. Dual ring-topology for media and network redundancy at host-level.

Hot-standby Transmitter Redundancy

Several transmitters can be connected to the same process point. The status associated with the measured values can then be used to filter out "Bad" measurements and select "Good." Both FOUNDATION Fieldbus and PROFIBUS PA use status bytes for each process variable. HART has a status bit in the "field device status" response byte that indicates the quality of the measurement as detected by the self-diagnostic algorithm in the device itself. Since the diagnostics is done in the transmitter, no additional comparison and alarm logic is required in the control strategy to reject bad values.

Make sure that you configure the control strategy to make use of this valuable status to select a "Good" measurement. This is easy using the standard selector block in the FOUNDATION Fieldbus programming language. The standard selector block handles three inputs but can be cascaded to handle more. A transmitter failure will cause one of the other transmitter readings to be selected, and a shutdown of the loop will therefore not be required. If three transmitters are used, all three have to fail before the selector output becomes "Bad," initiating a shutdown. Redundant transmitters thus result in fewer unnecessary shutdowns.

This is a classic example of fieldbus technologies making it easier to achieve measurement redundancy in order to achieve higher availability. In this example, three out of three (3oo3) sensors have to be degraded in order for a shutdown to occur. Use transmitters from diverse manufacturers to reduce the number of common cause failures. This is easy to do since the fieldbus technologies are interoperable between multiple vendors' products. If your plant adopts PROFIBUS PA technology, make sure to use a central controller that is able to incorporate the process variable status into the control strategy and manually configure equivalent interlocks. Similarly, if FOUNDATION Fieldbus devices are connected in a system that does not use the FOUNDATION programming language in the central controller, you must configure the logic manually.

Hot-standby Sensor Redundancy

The multivariable capability of fieldbus technologies allows transmitters to have dual inputs for connecting two sensors (figure 10-6). When high measurement availability is required mount these two sensors to measure the same point, and configure the control strategy within the device to select either "Good" of the two values. This is easy using the standard selector block in the

FOUNDATION Fieldbus programming language. A sensor failure such as a thermocouple burnout will cause the other sensor to be selected and therefore not necessitate a shutdown. Only if both sensors fail will the selector output be "Bad," thereby initiating a shutdown. Redundant sensors thus result in fewer unnecessary shutdowns. This is a classic configuration where two out of two (2oo2) sensors have to be degraded in order for a shutdown to occur.

If the controller does not support the FOUNDATION function blocks you must configure the logic manually. If FOUNDATION Fieldbus devices are used, it may therefore be a good idea to exclusively use the FOUNDATION Fieldbus function block diagram language, without mixing it with any intermediate proprietary languages.

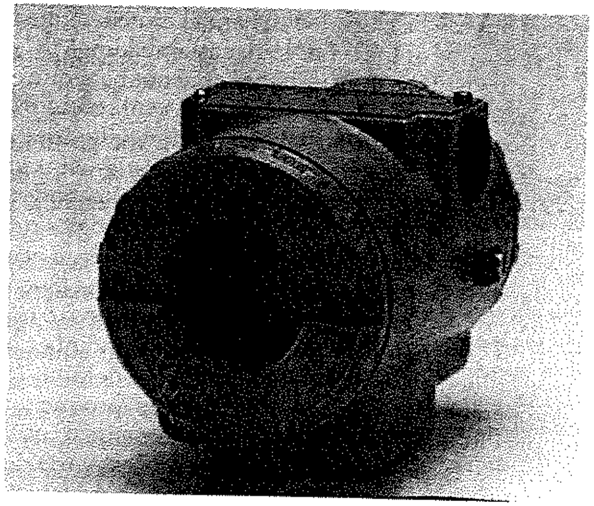


Figure 10-6. Dual input temperature transmitter with voting for safety or availability. (Courtesy of Smar)

Hot-standby Redundant Bus Power

Most field instruments require power from the bus to operate. If the power supply fails, the devices on the network and the associated loops will fail too. For higher fault tolerance it is therefore a good idea to use dual power supplies and dual power supply impedances. Use a power supply subsystem that ensures automatic and bumpless switchover from the primary to the secondary. Not all power supply impedances support redundancy, so it is advisable to check. Consider increasing availability further by reducing common causes, for example, by deriving the power from two different sources, such as, different phases, uninterruptible power supplies (UPSs), solar, battery, or others.

Hot-standby Linking Device Redundancy

For the operator to view the process and initiate actions the linking device or other interface must be functioning. The linking device is also the primary Link Active Scheduler (LAS) for the network. Should the linking device fail, the operators would be blind and the loops would all be shut down unless measures are implemented. Shutting down is safe but still not desirable because downtime means production loss. Therefore, it is a good idea to use redundant linking devices in a hot-standby arrangement together with redundant host-level networking. This will provide two complete independent communications paths routed separately and connecting the field-level network to the operators. In the event the primary linking devices fail, the secondary takes over, which ensures that plant floor data reaches the operator, providing him or her with a window into the process even in the presence of a fault.

Hot-standby Controller Redundancy

Usually, regulatory control for systems based on FOUNDATION Fieldbus is distributed into the field devices, eliminating the need for a central controller. For HART, PROFIBUS PA, and some instances of FOUNDATION Fieldbus, control is performed in shared centralized controllers that handle multiple loops. The loss of central control would have a serious impact on overall availability since it would affect many loops. Therefore, hot-standby redundancy is a necessity for centralized controllers that handle more than a few loops. A pair of controllers is configured with identical control strategies. Function blocks in the secondary controller are synchronized with those in the primary, which ensures a bumpless transition in the event of primary failure. It is prudent to centralize no more than thirty or so control loops in a single device.

Redundancy Separation

For redundancy to effectively increase availability, no nonredundant parts can be shared between the primary and secondary. Such parts make the system vulnerable to common mode failures. Redundant parts shall be physically, geographically, separated so stress affecting the primary does not affect the secondary and vice versa. For example, primary and secondary processors should slot into different backplanes separated by some distance. If the primary and secondary share a single common backplane, some backplane and other faults will likely affect both the primary and secondary. It may even be a good idea to mount the primary and secondary in separate panels. In other words, common single points of failures should be eliminated.

Similarly, when redundant host-level media or complete networks are used the two networks should be routed through different paths so neither is damaged by the same cause.

Backup Master

An active master device is required for communications to function. Without a master, the communication halts, loops shut down, and availability suffers. Since FOUNDATION Fieldbus typically performs control in the field it has an additional option to increase availability. The linking device on the HSE host-level network is usually the primary Link Active Scheduler (LAS) for the H1 field-level network, but if it fails another LAS takes over. If redundant linking devices are used, the secondary will take over the LAS role. You can use many FOUNDATION Fieldbus field instruments as link masters to assume the role of LAS in case the prevailing LAS fails (figure 10-7). In this way, communication and therefore control can continue without shutting down. In other words, you should enable backup LAS in a field device on networks that require high availability by configuring it as a link master device, particularly if redundant linking devices do not exist. For networks that emphasize safety it may be better not to configure any backup LAS. At design time, it may be a good idea to identify which networks benefit most from availability and which from safety and configure backup masters accordingly.

If the linking device has failed the operator has no visibility, but if a backup LAS in the field is used control still continues. Basically, the loops run on blind faith. It may be prudent to implement a procedure that limits the period of time a loop can be run in this way. If the communications cannot be reestablished the affected loops may have to be shut down.

Multiple Workstations

It may be a good idea to have at least two workstations, even for a small system. In the event one workstation fails, the operation can still be carried out from the second station.

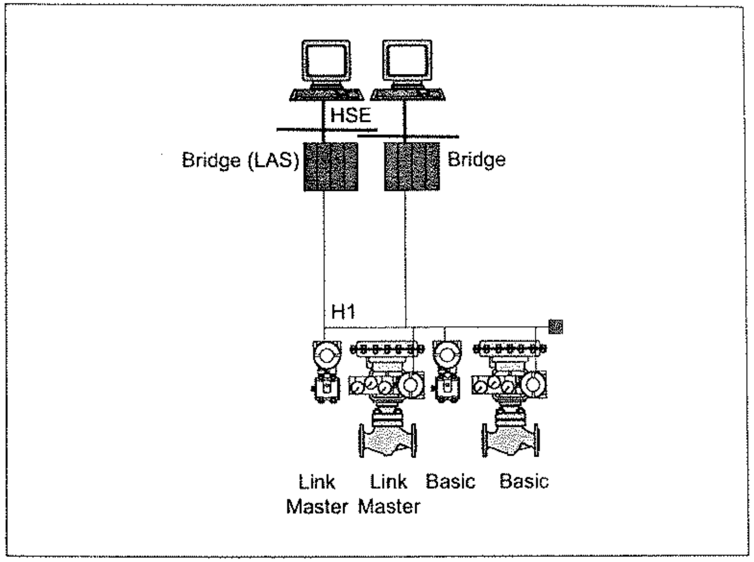


Figure 10-7. Consider backup LAS in the field, particularly if redundant linking devices are not available.

Availability through Diagnostics

Devices that are based on the fieldbus technologies have a wide diagnostic coverage built in. This self-diagnostic capability detects device failures, with different severity levels, and confirms that they are indeed genuine.

Reduce Mean Time to Repair (MTTR)

The longer a loop is not functional the worse the availability will be. An important way to increase availability is to detect problems early and fix them quickly, that is, to increase availability by reducing mean time to repair (MTTR). Make sure to configure the process visualization software so that if something is not right it displays status and sounds alarm to get the operator's attention. This will prompt operators to call on technicians to use the engineering tool to carry out detailed diagnostics that pinpoint the problem so it can be solved. Use diagnostics to verify if an abnormal condition is a process problem or an equipment problem. To use this scheme, the status must be communicated continuously to the operators, and the engineering tool must be permanently connected to access any device any time.

Reduce False Shutdowns

It is paramount that the self-diagnostics be reliable and accurate to avoid unnecessary spurious shutdowns—when what first appears to be a device fault really isn't. If many shutdowns occur downtime goes up and the availability of the system goes down. It is a good idea to utilize the fine granularity diagnostics provided by FOUNDATION Fieldbus and PROFIBUS PA technologies to distinguish between serious and less severe faults and take suitable action accordingly. In other words, internal fieldbus diagnostics is more precise than external discrepancy checking in comparing values from two transmitters. For example, when availability is a high priority shutdown should be configured so it only occurs for "Bad" quality, whereas for "Uncertain" just notifying the operator will do. Such configuration becomes particularly easy if you use the FOUNDATION Fieldbus function block programming language, which allows you to configure the desired behavior by using, for example, the status option (STATUS_OPTS) parameter. If you use HART or PROFIBUS PA devices, make sure to work with a central controller that supports the inclusion of status into the control strategy in order to configure interlocks based on the status.

Standby Health

Diagnostics is crucial if redundancy is to be effective. It is required if the fault is to be detected and if the switchover from primary to secondary is to occur. Diagnostics also tells the operator if one unit in the redundant pair is not functioning, meaning that a switchover for a second fault could not occur. This tells the operator to replace the faulty part of the redundant set so it is ready to take over at the next failure. Without information indicating that one failure has occurred redundancy is almost useless.

Additional Availability Measures

Fieldbus systems offer plants several other options and practices for increasing the fault tolerance.

Short-circuit Prevention

A network short circuit during routine maintenance would bring the devices down, which would make the final control elements shut down to their de-energized state. Measures should be taken to reduce the risk of a network short circuit. It is a good idea to use polarity-keyed, plug-in industrial connectors. These make it possible to connect and disconnect device wiring without incurring the risk of loose leads touching each other and short-circuiting the network.

An active junction box detects whether a spur branching out to one of the devices has been short-circuited. If it has, the junction box isolates it from the rest of the bus, thus preventing a prolonged short circuit of the entire network. In this way, not all devices and their associated loops on the network will be affected if a short circuit is caused by the improper handling of the wires. At design time, it may be a good idea to identify which networks require very high availability and to use active current limiting junction boxes on them.

Fewer Points of Failure

Use the capability of FOUNDATION Fieldbus to execute control in the field and thus eliminate the shared central controller device. If you don't use a controller for the loop, there is one point less that can fail, which means availability increases somewhat. Since there is no "main," there is no need for a redundant "backup" either. If the positioner executing the PID algorithm is working, control is working as well because the PID by itself cannot fail. It is merely a piece of software running on the same electronics and microprocessor that governs the rest of the device. What would stop the PID execution is a complete positioner failure, but without a positioner a central controller, even with redundancy, would be of no use either. In other words, since controls cannot work without a positioner the controller can be put in the positioner too.

Fail Operational

In the event of sensor, communication, or some other failure, you can make an output device either fail safely by shutting the loop down or fail operationally, in which case the valve would remain in its last position. When the valve is in its last position, the process is allowed to continue running, although it is not really being controlled since it is not moved to respond to changes.

Improving Safety

The fieldbus technologies do not really change the way process problems are detected and used to shut loops down. However, they can be used to increase the safety of a process control system. The most important feature is the diagnostics in the field instruments, which can be an integral part of the safety interlocks by shutting the loop down in case of instrument failure.

WARNING—The suggestions for safety and availability in this book are meant to be used for instruments in process control, not for shutting down risky processes where SIL-rated equipment has been deemed necessary. The plant may need both a process control and a shutdown system.

Safety through Diagnostics

Increased diagnostics means that more dangerous failures in the system are detected and fewer dangerous failures go undetected, particularly in the field instruments. Undetected faults are very dangerous because the process is not shut down but continues "unprotected." Make sure that you take advantage of the sophisticated instrument diagnostics and interlocks to improve system safety. Fieldbus detects additional shutdown scenarios, which makes it safer because it shuts down for failures that are not detected by conventional means.

Communication Failures

In blocks that are used in both PROFIBUS PA and FOUNDATION Fieldbus, the process I/O that is passed to and from function blocks and some related internal variables have both value and status. In addition to failure in sensor, actuator, or other device hardware a "Bad" status is also set in the event of communication error. PROFIBUS PA and FOUNDATION Fieldbus use different means to get values from and to the function blocks, and therefore communication error is determined differently as well. However, both operate on the basic principle that if there is no communication, that is, "no sign of life," the loop fails safe. In other words, no signal or wiring is required to initiate the fail safe; rather, fail safe is triggered by the absence of communication. Since a failure to communicate causes a shutdown, multidrop digital communication networks are very safe, without redundancy.

For PROFIBUS PA, the central controller reads input blocks in the field instruments cyclically. It is necessary to implement logic in the central controller to ensure that if this read is unsuccessful the controller puts the associated loop in manual or shuts it down by setting an "initiate fail safe" status to the corresponding positioner. The central controller also writes outputs cyclically to valve positioners, and so on. To fully explore the safety of the technology make sure to use a central controller with a programming language that provides easy access to status. Since the remote cascade setpoint input (RCAS_IN) is the main input for the AO block, fail safe gets activated if the remote cascade setpoint input does not receive successful communication from the central controller for longer than fail-safe timeout (FSAFE_TIME). Thus, communication failure or central controller breakdown cause a shutdown.

For FOUNDATION Fieldbus, the links between blocks in different devices are communicated at a very precise interval according to a time schedule. If anything in the chain from the transmitting device or field-level network fails, the intended receiving device will notice it immediately because the value is not received as scheduled. The input status will be set to "Bad - No communication." The programming language for the FOUNDATION Fieldbus function block diagram provides several options for bringing control to manual or shutting the loop down. The status works throughout the chain even if a bridge and host-level network fails. Thus, the shutdown will occur even if it is not the "local" network in which the valve fails but rather a value that should come from a failed remote network over a bridge.

Diagnostics instead of Discrepancy Checking

One of the practices that the fieldbus technologies change is the way in which transmitter failure is detected. In the past, transmitter failure was detected by the central controller comparing the reading between two analog transmitters connected to the same point. If a significant difference was detected between the two measurements one of the transmitters must have failed, and the loop had to be shut down. There was no way of telling which one had failed, what went wrong, or how severe the failure was.

Using the fieldbus technologies, however, the diagnostics that are obtained from the self-check inside a single transmitter indicate the validity of the measurement and shut the loop down in the event of a failure. In other words, diagnostics are done in the field instrument independently of the central controller. Every manufacturer has its proprietary ways of performing diagnostics, but the findings are presented in a standard way. Make sure you take advantage of the diagnostics capabilities of the fieldbus technologies to detect and act upon failures.

Instrument and Communication Failure Shutdown

The self-diagnostics provided in HART, FOUNDATION, and PROFIBUS PA devices detect instrument failures, not process problems, and they are therefore used for shutdown in the event of instrument failures. Process problems have to be detected using traditional means. Shutdown after process failure can be implemented in the process control system or in the safety-related system, if so required.

Both FOUNDATION Fieldbus and PROFIBUS PA report serious problems as "Bad" and some less severe measurement problems as "Uncertain." This allows interlocks to take suitable action. For example, when safety is important shutdown should occur both for "Bad" quality and for "Uncertain." It is a good idea to make full use of the capability diagnostics offers to automatically bring the process to a safe state in the event of instrument or communication failure. In other words, to improve safety the status should not only be displayed to the operator but also be part of the interlocks. Lost communication is indicated as "Bad" status and always results in loop shutdown.

Using trip interlocks for instrument or communication failure is particularly easy in systems that use the FOUNDATION Fieldbus control strategy programming language throughout the loop since these interlocks are built into the regular function blocks. These interlocks have to be manually configured for systems that are based on PROFIBUS PA or even FOUNDATION Fieldbus but use a different programming language.

In the FOUNDATION Fieldbus programming language problems like communication breakdown or device failure set a "Bad" status in the associated blocks. When the status is "Bad," the loop generally goes into manual mode. Alternatively, a fail-safe shutdown is initiated further down the control strategy. This is done by propagating the status using the integral shutdown path that is built into the standard function blocks, as explained in chapter 4, "Configuration." For example, in the PID block the mode will become manual to stop control. Optionally, the status to the valve positioner will become "Initiate Fail Safe," which will bring it to its predetermined fail-safe position. Since the diagnostics and shutdown mechanism is an integral part of the programming language of the FOUNDATION Fieldbus function blocks diagram it is a good idea to use a homogeneous system rather than a heterogeneous system. A homogeneous system is one that uses the FOUNDATION language in field instruments as well as the central controller without mixing in proprietary languages in which the status mechanism is not fully supported.

Plants can achieve similar shutdown for device failure or communication breakdown achieved even if the programming language of the FOUNDATION Fieldbus function block diagram is not used. Systems that use PROFIBUS PA or use FOUNDATION Fieldbus instruments but a proprietary central controller should have programming languages that have the capability to easily access the status for both read and write. This allows the user to manually configure the interlocks that are essential for safety.

Operator Notification

Failure of the control valve is a very serious condition because not only does it mean the valve is unable to control the process; it may also be unable to shut down properly if it is called upon to do so. For this reason, it is very important to use valve assembly diagnostics. The self-diagnostics in the valve positioner must be able to detect if the valve is stuck or has some other problem. To ensure that the shutdown channel in the process control system works, the host must be permanently connected to all field instruments, must continuously monitor for device failures, and must instantly report them to the operator so corrective action can be taken. It is a good idea to ensure that the system architecture allows devices to be monitored and maintained while control loops are running.

Configuration Correctness

Incorrectly configured interlocks may not work as intended. You can achieve greater safety by using the diagnostics built into the fieldbus devices rather than manually configuring comparisons. Similarly, you can improve safety if you use the fieldbus technologies to manage device configurations.

Device Configuration

A critical error that can be made with intelligent instruments after a replacement is to fail to configure all the parameters in the new device identically to the original device. Such a mistake is particularly easy to make if configuration is performed using a handheld that requires the user to remember to set all parameters. It is therefore a good idea to use an engineering tool in which the complete device configuration for all instruments is saved in a single database. Whenever a device is replaced, you can download in one go a complete configuration for the new device that is identical to that of the original device. This reduces the likelihood of configuration errors such as wrong range, damping, transfer function, channel, and the like. It is therefore important to use a configuration tool that stores all device parameter settings and changes in a database. This ensures that an identical configuration is downloaded after a replacement.

The simulation (SIMULATE) parameter in the I/O blocks has an associated warning in the block error (BLOCK_ERR) parameter to ensure that the device is not left in a simulation state. Moreover, the simulate parameter is nonvolatile, which ensures that it returns to normal operation after being powered off, as when it is moved after FAT.

In both FOUNDATION Fieldbus and PROFIBUS PA blocks, there is a static revision parameter (ST_REV) that counts the number of modifications. Make use of this parameter to trace any change.

HART and PROFIBUS PA use manual address assignment. Therefore, you should take care to ensure that the addresses for devices that are replaced are correct. This will prevent communication conflicts and downloading of the wrong configuration.

Interlock Configuration

Manually configuring logic for detecting instrument failures and shutting the loop down is a complex and error-prone process. To improve safety, it is advisable that you simplify by using the diagnostics and fail-safe mechanisms that are built into the FOUNDATION function blocks. The FOUNDATION function blocks are encapsulated program units with strictly defined data types that have been rigorously tested as part of the interoperability registration process. The programming language of the function block diagram has no global variables, thus eliminating complex interactions between program modules.

A mismatch in range and scaling may cause nonfunctional process interlocks to miss a process problem trip. Therefore, make use of the fact that fieldbus instruments operate in engineering units, not in percentage of range. To keep the control strategy homogeneous, do not mix in intermediate communication protocols and programming languages that introduce scaling or interrupt the status propagation.

If the configuration tool comes with a library of pretested control strategies it's a good idea to make use of them.

Safety through Decentralization

If shutdown interlocks are performed in a shared centralized controller, there is a small chance that a single undetected hardware failure may prevent the safety interlocks for several loops from functioning properly. By decentralizing this function plants can reduce the number of loops affected by an undetected fault.

Placing safety interlocks in the final control element has the additional advantage that the valve positioner can shut the loop down by itself independently of any centralized interlocks. In other words, even if the failure occurs in the controlling device or in the communication to the valve positioner it can still take proper action. This would not be possible using a conventional positioner or a positioner that does not have built-in safety interlocks. Use the fieldbus technologies to avoid actuation caused by "stale" or "crazy" values.

Other Safety Measures

The fieldbus technologies contain many subtle details that can be applied to improve the safety of a process control system. For example, you can reduce the time that the loops are in override mode using the self-diagnostics and remote partial stroke test for valves instead of doing manual testing. Self-checking occurs very much more frequently than manual checking. Selecting just one fieldbus technology is in itself a safety measure. Mixing multiple protocols on the plant floor would lead to translation problems and other safety concerns that can be avoided by using a single standard bus throughout the plant.

Safer Operation

In some applications, it is unsafe if the operator is unable to see what is occurring in the process. For nonhazardous loops, such as the water level in a tank, it may be OK to operate on "blind faith" for extended periods of time. However, for other loops the inability to view and initiate control actions even a brief moment would be unacceptable. Therefore, if the communication to the operators is lost, some loops must shut down whereas others could be allowed to continue. Consequently, for networks that have hazardous loops it may in fact be better not to have a backup LAS in the field since it may be safer to simply let the loop shut down automatically when the communication fails. This means that if the primary interface (and the secondary, if any) fails, leaving the operator blind, the field-level communication is also lost. This causes the output devices to fail safely, shutting the loops down. In other words, it may be a good idea at design time to identify which networks benefit most from availability and which benefit most from safety and then configure backup masters in the field accordingly. A compromise may be a plant policy that allows blind operation for a limited time, long enough to rectify the faulty interface or finish the batch. For outages that last any longer the loop should be shut down.

Fieldbus valve positioners with diagnostics and feedback are essential for safety since a loop that has a valve without diagnostics is unsafe no matter how good the rest of the system is. Indeed, without diagnostics in the valve the loop is being operated blindly because operators cannot see if the controller output comes though OK.

In the FOUNDATION Fieldbus programming language alarms and events are reported with priority levels 1 through 15. For consistency and to eliminate confusion, the same priority levels should be used for alarms throughout the system. That is, for safety reasons it is a good idea to use a host that homogeneously integrates alarming.

De-energize Safe

The shutdown strategies for any instrument failure or process problem rely on power to function. In the event that electric, pneumatic, or hydraulic power is lost the final control elements must fail safely anyway. This means that the mechanical design has to be such that the output returns to a safe position when power is lost. For example, a valve must be self-opening or self-closing.

WARNING—Make sure valves are selected with mechanical fail-open or fail-closed as an additional level of safety. This provides a last line of defense in case of power failure.

Discrepancy

The instrument diagnostics provided by fieldbus technologies is crucial for safety. However, detecting external problems may not be possible, for example, those occurring in the connection to the process. For example, there is no deterministic way for a transmitter to detect a blocked or damaged process connection. When it is necessary to detect faults external to the device two transmitters must be used. If two "Good" values deviate from each other, the problem is most likely external, such as buildup or "caking" on a pressure sensor diaphragm, that is, shutdown triggered by discrepancy checking for the piping.

Balancing Availability and Safety

Safety and availability are two conflicting goals. Safety means that loops will have to be shut down, stopping production and reducing availability. By striking the right balance for each loop, the plant's productivity and risk management can be optimized. The capacity to detect instrument failures and have options for handling them gives the user the ability to strike the right balance between safety and availability for each loop. Diagnostics is important both for safety and availability by making systems operate safely but with few spurious trips.

Uncertain

As we have seen, devices based on FOUNDATION Fieldbus and PROFIBUS PA distinguish between serious problems and less serious problems by indicating "Bad" and "Uncertain" status, respectively. This makes it possible to balance the interests of safety and availability. "Uncertain" quality includes conditions when the measured value is out of range or inaccurate for some reason. For inputs, it may additionally mean that the linking was changed or the value entered manually. A "Bad" value cannot be used for control and always shuts the loop down. However, a value that is "Uncertain" may still be useful, and it can therefore be used for loops that are not hazardous. However, "Uncertain" should cause a shutdown for a loop that is hazardous. This is easily achieved using the function blocks in the FOUNDATION Fieldbus programming language. In blocks for which status is relevant there is a status option (STATUS_OPTS) parameter where "Uncertain" can be preconfigured to be treated as "Good" for availability (merely notifying the operator) or as "Bad" for safety. This can be done for each individual loop. This optimizes the operation and allows two loops using the same value to act differently. One can be configured for availability (continuing to run), the other for safety (shutting down).

Where dual redundancy is used for availability the status is "Good" when both are working. It is "Uncertain.sub-normal" when only one is working and "Bad" when neither are working. This allows some loops to shut down while others continue.

The FOUNDATION function blocks also contain the status option (STATUS_OPTS) parameter, which is used to select whether a limited or manual value shall be "Uncertain" instead of the usual "Good." This enables some loops to shut down for safety reasons, while other loops using the same value may continue to operate.

Use the "Uncertain" status correctly to increase safety without losing fault tolerance. This is simple using the FOUNDATION programming language since the "Uncertain" status is built into the function blocks. However, by using controllers with proprietary programming languages increasing safety without sacrificing fault tolerance may still be achieved using manual configuration, provided the status is made available.

"Bad" Overrides "Uncertain"

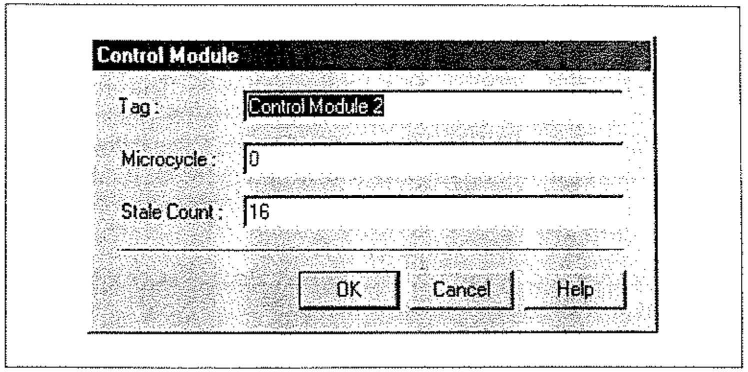
The "Bad" status has higher priority than "Uncertain." Thus, a value can first become "Uncertain" due to some minor problem and then become "Bad" due to a more serious problem. A small problem allows the loop to continue, whereas a subsequent major problem shuts it down. A communication failure in the loop must shut the loop down because it is not known what happens next. In the FOUNDATION Fieldbus programming language a communication failure sets the status as "Bad." In other words, a prolonged loss of communication always mean shutdown, since that is the only safe option.

Others

Prolonged communication failure should shut the loop down. Communication failures are detected in two ways. They can be detected by using a timeout, in which case the loop is shut down if the communication has not been working within a predetermined period of time. They can also be detected if a preconfigured number of communication attempts have failed. For example, in the FOUNDATION Fieldbus programming language you can configure the number of macro cycles that can be missed before the function block link is considered "stale" (figure 10-8). To reduce the number of unnecessary trips the communication timeout and stale counts can be configured such that a short "glitch" in the communication does not shut the loop down. The stale count limit can typically be configured so as to tailor the behavior of the function block links in an individual control module. For example, in a valve positioner you can set how long the process variable communication can be lost before it shuts the loop down. Set it low to be safe and long for fault tolerance.

Configuring Function Blocks

In the FOUNDATION Fieldbus function block programming language there are parameters that at design time allow the interests of safety and availability to be balanced with respect to failures.



Tag: Control Module 2

Microcycle: 0

Stale Count: 16

Figure 10-8. Common stale count limit setting for all links in a control module.

Fail Safe

The status option (STATUS_OPTS) parameter in the FOUNDATION function blocks provides options for the way communication faults in the upstream part of a control loop shall be propagated downward. Using the option "IFS if bad IN," you can select whether a failed transmitter shall shut the loop down safely or merely put it in manual, thus letting the process run. Using the similar option "IFS if bad CAS_IN," you can configure a secondary PID in a cascade to fail safe or continue in automatic when the remote setpoint fails. In case of automatic, the PID will operate with a fixed setpoint in a basic PID scheme. This scheme sets failure in the primary part of the loop on one fault but allows secondary loop to continue in a simpler mode.

Fault Recovery

There are a number of options for deciding what should happen after a communication failure of a value that comes from a non-fieldbus device recovers after the problem has been fixed. An affected control loop can either resume control immediately to achieve higher availability or, more cautiously, it can remain "failed" until the operator unlocks it. For the shed option (SHED_OPT) parameter the option "No Return" provides more safety whereas the option "Normal Return" provides greater availability. For the status options (STATUS_OPTS) parameter you can provide greater safety by using the "Target to ..." options, whereas by not enabling them you can provide greater availability for function block links between FOUNDATION devices.

Limited and Manual Values

In the FOUNDATION function blocks, there is a status option (STATUS_OPTS) to show limited values as "Uncertain" or even "Bad" instead of the default "Good." Similarly, manually entered values can also be shown as "Uncertain." Thus, downstream blocks can optionally take safe action in these situations.

Summary Table

The various options for striking a balance between safety and availability are summarized in table 10-1. Note that for a majority of the options the default setting is aimed toward process control availability rather than safety.

Table 10-1. Options for balancing the interests of safety and availability

Feature	Safety	Availability
STATUS_OPTS	Don't (i.e. as "Bad")	Use "Uncertain" as "Good"
STATUS_OPTS	Target to Manual if "Bad" IN	Don't (i.e. return to normal once OK)
STATUS_OPTS	Initiate fault state if IN is "Bad"	Don't (i.e. merely go manual)
STATUS_OPTS	Initiate fault state if CAS_IN is "Bad"	Don't (i.e. merely go automatic)
STATUS_OPTS	Set Target mode to manual if IN is "Bad"	Don't (i.e. return to normal once OK)
STATUS_OPTS	Set quality as "Uncertain" if limited	Don't (i.e. consider "Good")
STATUS_OPTS	Set quality as "Bad" if limited	Don't (i.e. consider "Good")
STATUS_OPTS	Set quality as "Uncertain" if block is manual mode	Don't (i.e. consider "Good")
IO_OPTS	Fault State to value	Don't (i.e. freeze output)
IO_OPTS	Use Fault State value on restart	Don't (restart from present position if available)
Function block link stale count limit.	Few	Many
SHED_OPTS	No return	Normal return
FSAFE_TIME	Short	Long
FEATURE_SEL	Fault state supported	Don't (i.e. disabled)
Backup LAS function	Disabled	Enabled
SHED_RCAS	Short	Long
SHED_ROUT	Short	Long

EXERCISES

10.1 Is safety and availability the same thing?

10.2 Does diagnostics improve availability?

10.3 Does diagnostics improve safety?

10.4 Does redundancy increase safety?

10.5 Is a loss of communication unsafe?

10.6 How does the "Uncertain" make it possible to balance the interests of safety and availability?

10.7 Does a backup LAS improve safety?

BIBLIOGRAPHY

1. Faller, Rainer, *Proposal for an Interim Solution for the Use of Non-safety-related Intelligent Field Instruments in Safety Applications in the Process Industry*, rev 3.2. TÜV Product Service, 7 October 1998.
2. Goble, W. M., *Control System Safety Evaluation and Reliability*, 2d ed., ISA - The Instrumentation, Systems, and Automation Society, ISBN 1-55617-636-8, 1988.
3. Gruhn, Paul, and Harry L. Cheddie, *Safety Issues Related to Field Devices*, volume EMC 17.01, ISA, 2000.
4. ANSI/ISA-84.01-1996, *Application of Safety Instrumented Systems for the Process Industries*, Research Triangle Park, NC: ISA, ISBN 1-55617-590-6, 1996.

11

How Fieldbuses Work

This chapter is for the "propeller heads" who want to know a little bit more about how the fieldbus technologies work. However, you don't have to understand how the fieldbus technologies work internally to use them. These protocols are complex, but manufacturers typically implement them in systems in such a way that they appear easy. Most readers may therefore skip this chapter or read it merely out of curiosity. The main focus is the field-level networks, that is, HART, FOUNDATION H1, and PROFIBUS PA. There are plenty of books describing how Ethernet, IP, TCP, and UDP work. These subjects are therefore not covered here.

Primer

HART, FOUNDATION Fieldbus, and PROFIBUS are all serial communication protocols. Information is converted into a stream of ones and zeroes, which are transmitted over the media. These technologies are half-duplex, meaning that communication is bidirectional—that is, a device can both transmit and receive, but not at the same time.

OSI Model

HART, FOUNDATION Fieldbus, and PROFIBUS are all based on the Open System Interconnection (OSI) reference model as defined in the ISO 7498 standard. Messages are parsed through these layers, with each layer performing a set of functions. Layer 1 includes the physical media the message travels on, typically a wire. Above layer 7 is either the "real" device function, such as measurement, actuation, or control, or the operator interface in a host. A message makes its way down through the layers in the transmitting device, over the wires, and then up through layers in the receiving device. If the message is a request, the device attends to it and responds by passing a message back the opposite way.

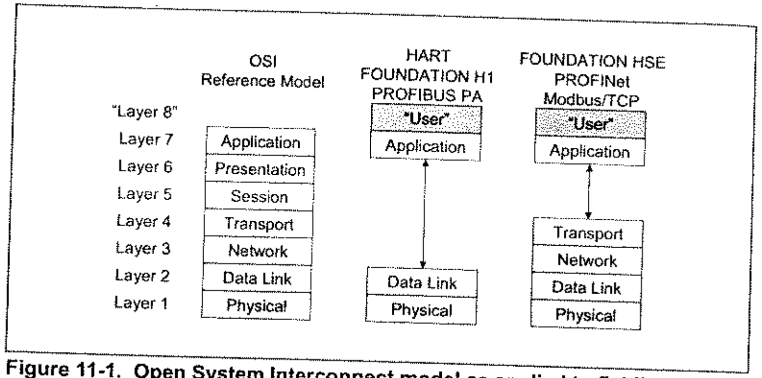


Figure 11-1. Open System Interconnect model as applied to fieldbus technologies.

Typically, each layer in the transmitting device adds a piece of information to the original message, which is then stripped off in the corresponding layer of the receiving device. For example, the data link layer may add a destination address that is then removed in the data link layer of the receiving device.

Most protocols do not implement all seven layers. For field-level protocol, usually only levels 1, 2, and 7 are implemented (figure 11-1). For host-level protocols, levels 3 and 4 are usually also included. Both Foundation HSE and PROFInet use Ethernet for layers 1 and 2. The Internet Protocol (IP) is the protocol for layer 3. PROFInet and Foundation HSE use the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), respectively, for layer 4. The network administrator in your company will surely have a book describing the function of Ethernet, IP, TCP, and UDP.

Layer 7 Interoperability

For devices to be able to interoperate with each other on the same network all the seven OSI layers and the user layer must be the same. Merely using the same physical layer does not provide any interoperability; it just guarantees that devices are electrically compatible. In other words, using RS232, RS485, Ethernet, and so on is not sufficient for interoperability, nor is TCP/IP. In fact, most protocols using these physical layers are proprietary. The most important aspect of the HART, PROFIBUS PA, and Foundation technologies is that they specify in detail the application and user layers that are required for interoperability. This sets them apart from many other protocols. These fieldbus protocols specify not only how data is transported but also its semantics, that is, what the information means. For example, all HART devices use the same codes for engineering units, and all Foundation control loops are put into manual mode the same way. Without application- layer and user-layer standards plants will ultimately require custom programming and parameter mapping to make the system work.

Physical Layer

The physical layer defines how the signal is physically transmitted over media from one device to another, electrical or otherwise. RS232 and RS485 are classical examples of physical-layer standards. Note, however, that these are not complete protocols. The physical layer does not interpret the data; it just passes it to and from the data link layer. The installation rules are described in detail in chapter 3, "Installation and Commissioning."

HART

The most important aspect of HART is that it retains the analog 4-20 mA signal. Digital communication at the rate of 1200 bit/s is superimposed on the same two wires as the analog current loop signal without disturbing it.

HART Encoding

HART uses frequency shift keying (FSK) according to the Bell 202 standard to superimpose a digital signal as AC on top of the 4-20 mA DC. Bits that are "0" and "1" are encoded as sine waves of the frequencies 2,200 Hz and 1,200 Hz, respectively. When transmitting, the bits are converted to the corresponding frequency, and when receiving the frequencies are converted back to bits of the corresponding state.

The frequencies are sinusoidal and completely symmetric and therefore add no DC component. Moreover, the frequencies are removed by low-pass filters on the inputs of analog devices in the current loop. Thus, digital communication can be done without disturbing the analog 4-20 mA signal. A HART communications chip (figure 11-2) performs the signal modulation and demodulation.

Figure 11-2. HART communications controller. (Courtesy of Smar Research Corporation)

HART Signaling

The FSK signal is transmitted by modulating the current in the loop with an amplitude of approximately 0.5 mA (figure 11-3). Passive devices, such as most field instruments, do this essentially by rapidly altering their current consumption. Active devices like a handheld do it by injecting a signal.

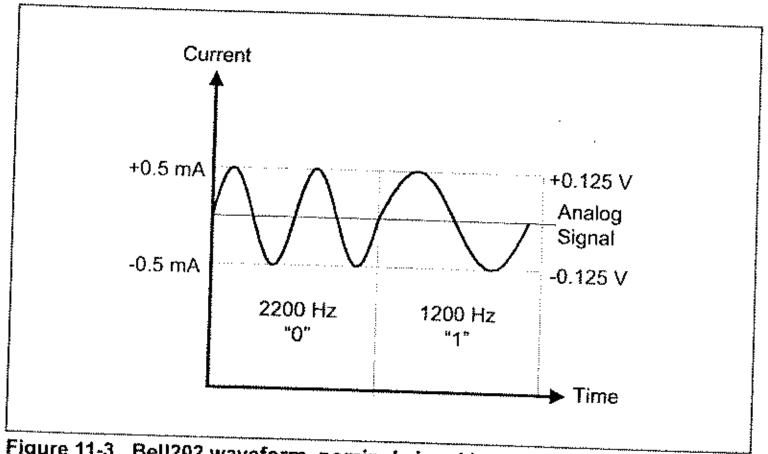


Figure 11-3. Bell202 waveform, nominal signal levels.

The HART network must have some resistance between the devices and the power supply. This resistance, usually the 250 ohm shunt on the host input module, has two functions: it prevents the DC power supply from short-circuiting the AC communication signal, and it also acts as a shunt for the communication signal. As the transmitted FSK current passes through the shunt a 0.125 VAC voltage drop is created. All devices on the network pick up this AC voltage and are sensitive enough to receive even if there is attenuation of the signal along the wires. That is, transmission is done as a current, reception as a voltage. If adequate resistance does not exist in the loop the voltage becomes too small to detect and communication will not work. In multidrop mode, several transmitters are connected in parallel, typically with the 4-20 mA disabled and the output fixed to about 4 mA, independent of input.

HART Character

HART uses an asynchronous mode of communication, meaning that data is transmitted without a clock signal. To keep the transmitting and receiving devices synchronized, data is transmitted one byte at a time. It is led by a start bit "0" and followed by the eight bits of the real data, one parity bit which is odd and one stop "1" (figure 11-4). The parity is set either "0" or "1," which makes the total number of 1s in the data and parity together odd. The parity bit provides additional data integrity within the received character by checking that the total number of 1s is indeed odd.

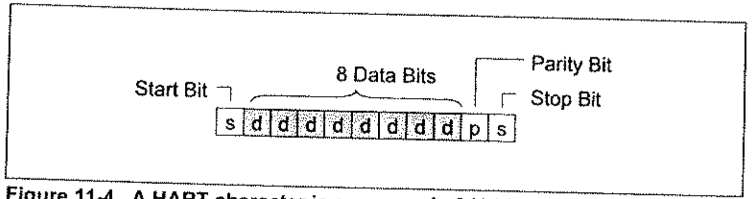


Figure 11-4. A HART character is composed of 11 bits.

IEC 61158-2 and ISA-50.02, Part 2-1992 Fieldbus Standard for Use in Industrial Control Systems. Part 2: Physical Layer Specification and Service Definition (FOUNDATION Fieldbus H1 and PROFIBUS PA)

Digital communication at the rate of 31.25 kbit/s is superimposed on the same two wires as the device power.

IEC Physical Layer Frame

The frame has a preamble that is transmitted at the beginning of each message to "wake up" all other devices on the network and allow them to synchronize their receiver to the transmitting device. A start delimiter denotes the end of the preamble and the beginning of the actual message, coming to or from the data link layer, after which there is an end delimiter (figure 11-5).

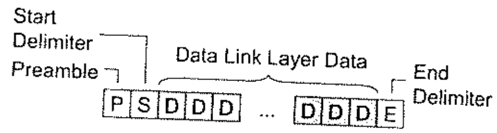


Figure 11-5. IEC physical layer frame.

IEC Encoding

IEC 61158-2 is synchronous, using Manchester biphase-L code that mixes a clock signal together with the data in a single signal. Because the signal is self-clocking, no start and stop bits are required. This enables frames to be transmitted in a single continuous data stream. Data bits that are "0" and "1" are encoded as rising and falling transitions, respectively (figure 11-6). When transmitting, the bits are converted to the corresponding transition, and when receiving, the transitions are converted back into bits of the corresponding state. The Manchester code is completely symmetric and therefore adds no DC component.

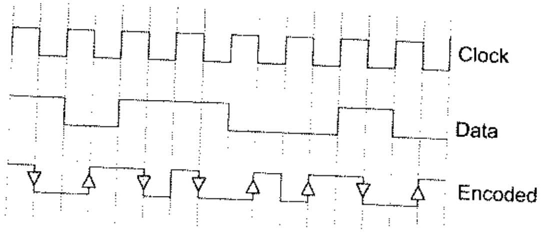


Figure 11-6. Manchester biphase-L encoding.

The start delimiter and end delimiter are encoded, including special unmistakable nondata transitions (figure 11-7).

A fieldbus communication controller (figure 11-8) performs the signal encoding and decoding, as well as many other functions.

IEC Signaling

By modulating the current with amplitude of approximately 10 mA, the Manchester signal is transmitted over the wire media (figure 11-9). Passive devices, such as most field instruments, do this essentially by rapidly altering their current consumption. Active devices like a host interface do this by injecting a signal.

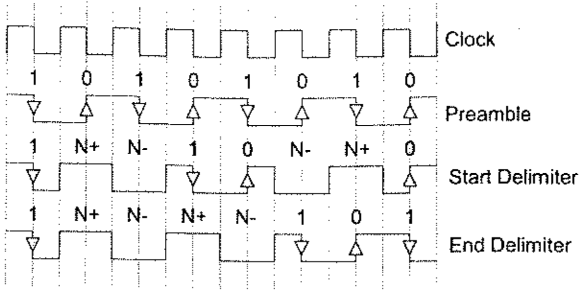


Figure 11-7. IEC codes for preamble and delimiters

Figure 11-8. IEC 61158-2 chip used for both FOUNDATION H1 Fieldbus & PROFIBUS PA. (Courtesy of Smar Research Corporation)

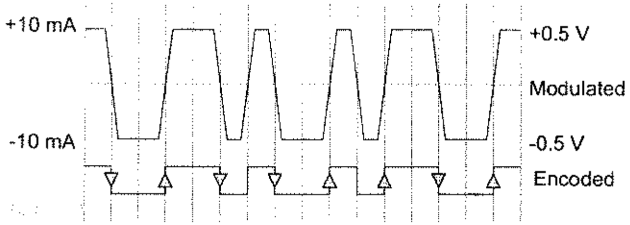


Figure 11-9. Transmitted waveform, nominal signal levels.

The network has an impedance module located between the devices and the power supply. This impedance prevents the DC power supply from short-circuiting the AC communication signal. The network also has a 100 ohm terminator in each end that acts as a shunt for the communication signal (figure 11-10). The parallel shunts result in a 50 ohm impedance, and as the transmitter's Manchester current passes through it creates a 0.5 VAC voltage drop. All devices on the network pick up this AC voltage and are sensitive enough to receive even if there is attenuation of the signal along the wires. In other words, transmission is done as a current, reception as a voltage.

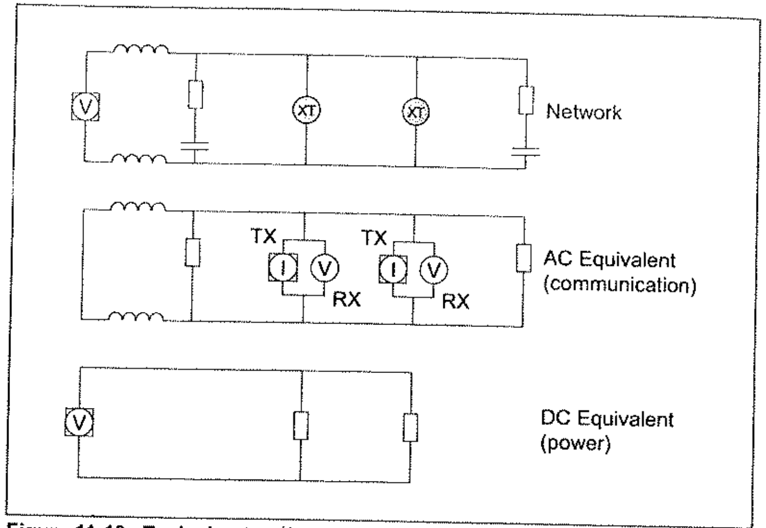


Figure 11-10. Typical network and its AC and DC equivalents.

At the same time, the terminators also prevent reflection when the signal travels down the network and reaches the end of the wires.

Data Link Layer

The data link layer controls when and for how long a device can gain access to the network in order to avoid conflict between two or more devices that want to transmit at the same time. The data link layer also includes addressing, which ensures that the message reaches its intended recipient. Lastly, the data link layer is responsible for error checking. The data link layer does not interpret the data; it merely passes it through between the physical and upper layers.

HART

The HART protocol is a quite simple protocol, primarily of a master/slave nature.

HART Device Types

The HART data link layer recognizes three device types:

- Master
- Slave
- Burst mode slave

The primary master is typically a system host, the secondary master a handheld configuration tool, and the slave some form of field instrument, such as a transmitter or valve positioner. Master devices initiate the communications whereas the slaves merely respond to commands.

HART Addressing

HART has two addressing schemes: polling address and unique identifier. Every HART device has a five-byte unique identifier, which is a hardware address that consists of a one-byte manufacturer code, one-byte device type code, and a sequential number. The identifier thus uniquely distinguishes the device from among all others in the world. The master uses this unique ID, also known as its "long address," when it communicates with the slave. The HART Communication Foundation universally administers the manufacturer code so as to eliminate the potential for duplicated addresses. The manufacturer assigns a device type code and sequential number.

The single-byte polling address, also known as the "short address," ranges from 0-15, where 0 means single-unit mode and 1-15 means multidrop mode. The polling address is only used with very old HART devices that do not support the long address format and when first establishing communication with an unknown device.

Typically, HART is used in single mode, in which case the master only has to poll address 0 to get the unique ID from the slave. In multidrop mode, the master typically checks all polling addresses 1-15 to verify if a device is present. It can then presents the user with a list of live devices on the network. Alternatively, the user can enter the tag of the desired device, and the master will broadcast it to get a response containing the unique ID from the slave with that tag. Both the polling address and unique ID indicate whether the message is being exchanged with a primary or secondary master and whether the slave is in burst mode.

HART Arbitration

HART has two modes of operation: master/slave and burst. In master/slave mode, a master will request a slave, which will respond to the command from the master. The HART protocol allows two masters on the network: a primary master, which is typically a central controller, and a secondary master, which is typically a handheld terminal. Arbitration between the two masters is based on timing. Slaves in master/slave mode do not initiate communication. A slave only responds to master requests.

Slave burst mode is initiated by a command from the master. The slave will continuously broadcast the command response until a master instructs it to stop. In other words, burst-mode responses are generated without corresponding request frames. Using the burst mode, the data will be updated much faster since the slave transmits without waiting for a master request.

Collision on the network is avoided by first detecting if any other device is transmitting before a frame is transmitted. Timers control access sharing between primary masters, secondary masters, slaves, and slaves in burst mode. The two masters have the same priority for accessing the bus to initiate communication. A master that has just transmitted a message must wait longer than the other master before it can access the bus. Thus, if both masters are accessing the bus they will alternate. Slaves do not initiate communication; they simply respond to requests and must do so within a limited time. Slaves in burst mode wait longer to transmit than does a master, which allows the master to instruct the burst-mode device to stop.

HART Frame Formats

The HART message frame is made up of nine parts (figure 11-11), which are described later in this section. The response from the slave includes command response code and field device status information that is not included in a request message. Usually, there is a short idle period between the transmission of each character.

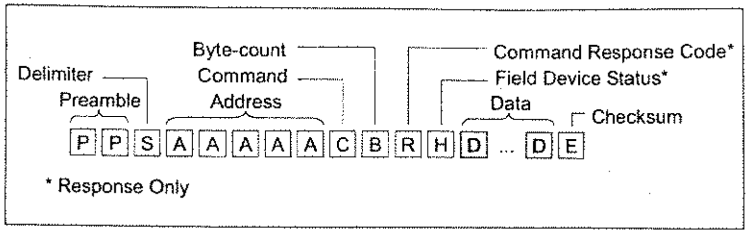


Figure 11-11. Characters in HART frames.

The *preamble* consists of two bytes that are first transmitted at the beginning of each message to "wake up" all other devices on the network and allow them to synchronize their receiver to the transmitting device.

The *start character* is a single byte that denotes the end of the preamble and the beginning of the actual message. The value of the start character tells if the frame is a request from a master, a response from a slave, or a request from a slave in burst mode. The start character also tells if polling address or unique ID is used for addressing.

The *address* is five bytes when the long unique ID address is used or one byte when the polling address is used.

The *command* is a single byte that represents the HART command associated with the message. No interpretation of the command is performed in the data link layer. The command and data is merely passed through to and from the application layer.

The *byte count* is a single byte that indicates how many more bytes there are until the end of the message, excluding the checksum. Using this information, the receiver can verify when it has reached the end of the frame.

The *response code* is a single byte that is included only in response messages. If the message was not received successfully, the response code from the slave will indicate the form of communication error it encountered. If the message was successfully received, the response code will instead indicate if the slave was able to successfully execute the command. If there were any problems executing the command this will be indicated.

The *field device status* is a single byte that is included only in response messages to indicate the health of the device. See chapter 9, "Maintenance and Asset Management."

The *data* is the information in a write request or in the response to a read request. No interpretation of the data is performed in the data link layer. The command and data is merely passed through to and from the application layer. The number of bytes can be anywhere from none up to twenty-four.

The *checksum* is a single byte of longitudinal Cyclic Redundancy Check (CRC). The transmitting device generates it by doing an XOR (a logic exclusive-or) of all the bytes in the message from the start character. The same function is likewise performed in the receiving device, and the result must match. The checksum thus provides data integrity by detecting corrupted messages.

FOUNDATION Fieldbus H1

In FOUNDATION Fieldbus, any device can initiate communication as long as it holds the right to do so. Transmission on the FOUNDATION H1 network is controlled by the link active scheduler (LAS). The data link layer appends five to fifteen bytes of control information at the beginning of the message and two bytes of error check at the end when it transmits. It removes the information when receiving.

H1 Device Types

The FOUNDATION H1 data link layer recognizes three device types:

- Basic
- Link Master
- Bridge

Link masters are able to become LAS, whereas basic devices are unable to become the LAS. Field instruments like transmitters, and valve positioners are typically basic devices, whereas host interfaces are link masters or bridges. However, many field devices can also be configured as link masters and are able to take over the LAS role.

H1 Addressing

The FOUNDATION H1 data link layer uses a single-byte network address. Addresses 0-15 are reserved for internal functions, 16-247 are used for instruments, 248-251 are default addresses for uninitialized devices, and 252-255 are used by temporarily connected devices such as handheld devices. The address is assigned automatically by the LAS when the device is connected to the network. The automatic address assignment eliminates the risk of address duplication.

H1 Arbitration

As far as the FOUNDATION data link layer is concerned there are two types of communications:

- Scheduled communication (foreground traffic)
- Unscheduled communication (background traffic)

Data that only has to be communicated infrequently is transmitted acyclically using unscheduled communications. Unscheduled communication includes, for example, the host reading and writing parameters in the field instruments. The LAS passes a token between the devices using the Pass Token (PT) message. Once a device holds the token it can send messages for up to the maximum token hold time or until it is finished, whichever is shorter.

Data that has to be communicated cyclically on a precise periodic basis is transmitted using scheduled communication. Scheduled communication includes, for example, function block links between devices. The LAS has a schedule determining when the cyclic values in the devices on the network shall be transmitted. When a value is scheduled for transmission, the LAS transmits a Compel Data (CD) message to the device, which subsequently publishes the data on the network using a broadcast. Devices that subscribe to the published value all receive it at the same time.

Scheduled communication has the highest priority on the network. After all scheduled data has been published, the time remaining before the next macro cycle is used for unscheduled communication and some other functions. The LAS maintains the live list of known devices on the network. A Probe Node (PN) message is periodically sent to the unused addresses to detect if any new device has been added. When receiving such a message, a device sends a Probe Response (PR) message to let the LAS know it is present, and the device gets added to the live list. The LAS then broadcasts the updated live list to all devices on the network. The LAS also ensures that the clocks in the devices on the network are precisely synchronized by periodically broadcasting a Time Distribution (TD) message. The time distribution ensures that function blocks to be executed and messages to be published are synchronized among all devices on the network.

The LAS schedule and token-passing mechanism ensures that only one device at a time is granted permission to access the bus, thereby avoiding collisions. The LAS ensures that the scheduled data is published and that all devices have a chance to communicate. The timing of scheduled communications is precisely defined and therefore inherently deterministic.

H1 Error Detection

Communication errors are detected by transmission of a Frame Check Sequence (FCS) in the message generated by the transmitting device. In the receiving device the received FCS is checked against internally computed FCS.

PROFIBUS PA

In PROFIBUS PA, only a master device can initiate communication. The data link layer ensures that only one device at a time accesses the bus. The data link layer also handles addressing and error check.

PA Device Types

The PROFIBUS data link layer recognizes two device types:

- Master
- Slave

Masters include central controllers and a host configuration tool. A slave is some form of field instrument, such as a transmitter or valve positioner. Master devices initiate the communications, whereas the slaves merely respond.

PA Addressing

All addresses up to 125 are individual addresses that are available for devices. Address 126 is a default address. The lower the address, the higher the priority. Therefore, the lower addresses are typically occupied by masters.

PA Arbitration

The masters pass a token between themselves in a logical ring in address sequence, using a special message (figure 11-12). Once a master receives the token, it can send messages. The time it takes the token to make a complete circle to all masters is called token rotation time. The master that holds the token can poll slave devices, which will respond. The token-passing mechanism ensures that only one device at the time is granted permission to access the bus. This prevents collisions.

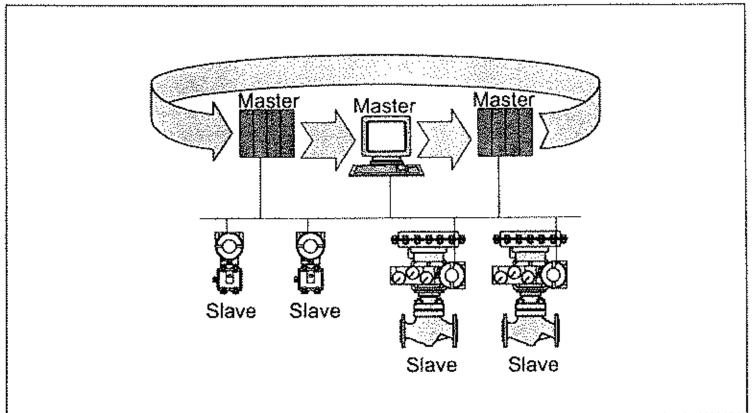


Figure 11-12. Masters pass token from one to the next in a logical ring.

The token mechanism is free running, meaning that there is no synchronization between the communication and the user layer functions. For that reason, the token rotation time should be targeted as low as possible to ensure that inputs and outputs are updated with minimal delay. The token-passing procedure ensures that each master will have enough time to fulfill its communication tasks if a sufficient token rotation time is configured.

Application Layer

The application layer is where simple data types and more sophisticated objects are defined. Several functions are provided for accessing these. Because interoperability is not just transferring bits and bytes from one place to another, the application layer is essential to making devices interoperable.

The HART application layer differs from the FOUNDATION application layer in that HART provides very direct commands that are specifically targeted toward a function like calibration or reset. FOUNDATION and PROFIBUS PA, on the other hand, have generic services that read and write all kinds of parameters, some of which invoke functions like calibration and reset.

PROFIBUS PA does not specify an application layer. However, the functions performed by the PROFIBUS DP communications profile upon which PA is based are very similar to those of an application layer and are therefore included in this section.

HART

The HART application layer is based on a set of commands for accessing functions and data in field instruments.

HART Data Types

HART supports a few basic data types: IEEE 754 float for values that are analog in nature; ASCII and packed ASCII for strings; and unsigned integers of up to 1, 2, or 3 bytes used, for example, for enumerated options. The options that are associated with universal and common practice commands as well as some others are standardized in common tables. These include, for example, engineering unit codes.

HART Variables

Measured and calculated variables in the device are referred to as dynamic variables. Four of these are fixed or can be assigned as dynamic variables: primary variables, secondary variables, tertiary variables, and fourth variables. The primary variable corresponds to analog output number 1. If additional outputs exist these correspond to the secondary variable and so on.

HART Commands

Masters read and write information in slave devices and invoke functions using commands. Every command has a different function and different amount of data associated with it. There are three classes of HART commands:

- Universal
- Common practice
- Device-specific

The universal and common practice commands are part of the HART standard. The universal commands are mandatory and supported by all HART devices. On the other hand, the common practice commands are optional, and each device typically only supports a few of them as applicable. Most functionality in a device is accessed using the universal and common practice commands. Additional less common functions in the device are accessed using the device-specific commands. A master transmits a request to read or write values to a slave, which will return a response.

The universal commands are used to identify the device in order to establish communication. Universal commands can be used to monitor and read basic device configuration. The universal commands are as follows:

- Read Unique Identifier
- Read Primary Variable
- Read Primary Variable Current and Percent of Range
- Read Dynamic Variables and PV Current
- Write Polling Address
- Read Unique Identifier Associated with Tag
- Read Message
- Read Tag, Descriptor, Date
- Read Primary Variable Sensor Information
- Read Primary Variable Output Information
- Read Final Assembly Number
- Write Message
- Write Tag, Descriptor, Date
- Write Final Assembly Number

The *common practice commands* write the basic device configuration, and some essential functions can be performed as well. Some common practice commands are the following:

- Read Transmitter Variables
- Write Damping Value
- Write Range Values
- Set Lower Range Value
- Set Upper Range Value
- Loop Test
- Perform Self-test
- Reset
- Input Zero Trim (Calibration)

- Write PV Units
- Current Zero Trim (Calibration)
- Current Span Trim (Calibration)
- Write Transfer Function
- Write Sensor Serial Number

For example, command #14, "Read primary variable sensor information," requires no request data but has sixteen bytes of response data (figure 11-13).

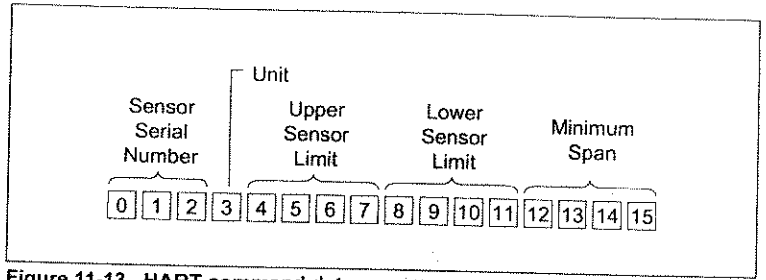


Figure 11-13. HART command data.

A slave response includes a code notifying the operator if the requested command was executed successfully. The response may be that the command was executed successfully, is not implemented, the device is too busy, or that single or multiple errors or warnings were encountered.

The device status indicates the health of the device. This is explained in chapter 6, "Troubleshooting."

FOUNDATION Fieldbus

The FOUNDATION application layer consists of two sublayers: the Fieldbus Access Sublayer (FAS) and Fieldbus Message Specification (FMS). As the message makes its way from the function blocks each layer adds control information to it, which is stripped away at the corresponding layer in the receiving device (figure 11-14).

FOUNDATION Data Types

Thirteen different simple data types are defined: Boolean, integer, unsigned, floating point, visible string, octet string, date, time of day, time difference, bit string, time value, null, and packed. These are combined to form the more complex data objects.

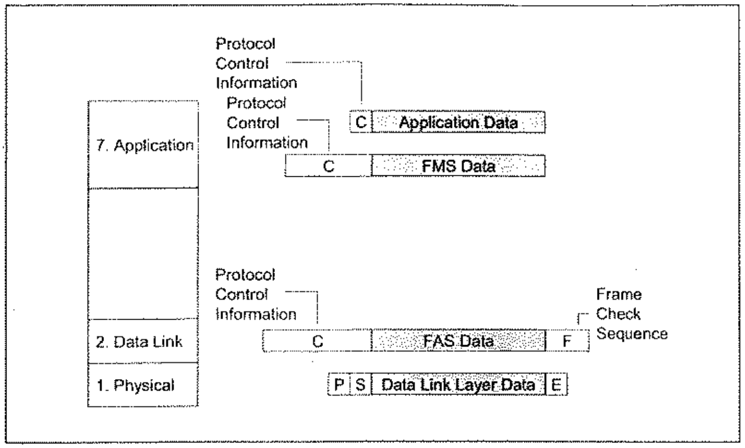


Figure 11-14. FOUNDATION H1 frame construction.

FOUNDATION Communication Relationships

In the FAS, Virtual Communication Relationships (VCRs) map to scheduled and unscheduled communication services in the data link layer. A typical device would use a few of each of the three types of VCR:

The *publisher/subscriber VCR (BNU)* is used for cyclic publishing of function block outputs that are subscribed to by function block inputs. Publisher/subscriber communication is buffered, meaning that when the function block generates a new output value it overwrites the old. Publisher/subscriber communication is scheduled and one-to-many, meaning that the value is broadcast to many subscribers simultaneously (figure 11-15). Publisher/subscriber communication need not go through a central host; it can be peer to peer directly between devices in the field.

The *report distribution VCR (QUU)* is used for the acyclic transmission of trends, alarms, and event notifications to the host. Report distribution is queued, meaning that when the function block generates an alert it is sent in an order determined by priority and time of occurrence and does not overwrite earlier alerts. Report distributions are unscheduled and one-to-many, meaning the value is broadcast to many subscribers simultaneously (figure 11-16).

The *client/server VCR (QUB)* is used, for example, for acyclic reading and writing of device parameters, downloading configurations, and other activities from the host. Client/server communication is queued, meaning that requests are sent in an

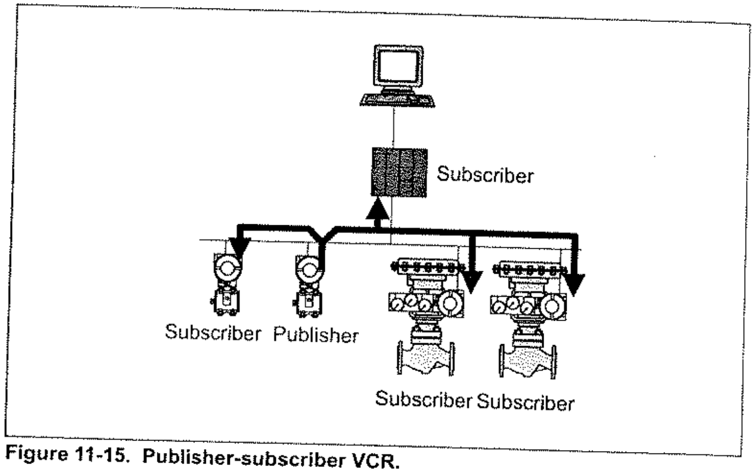


Figure 11-15. Publisher-subscriber VCR.

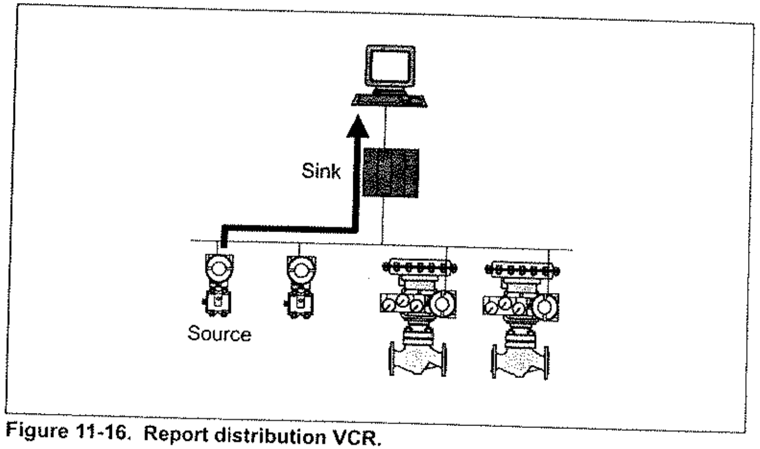


Figure 11-16. Report distribution VCR.

order based on priority and time of request without overwriting earlier requests. Client/server communication is unscheduled and one-to-one, meaning the value is sent to only one destination (figure 11-17).

FOUNDATION Object Dictionary (OD)

The FOUNDATION Fieldbus protocol is object oriented. The information in the devices is accessed in the form of objects. In FOUNDATION FMS, the objects for configuring device and strategy in a node are listed in an object dictionary (OD). Each object is identified by an index. For example, every function block and every parameter

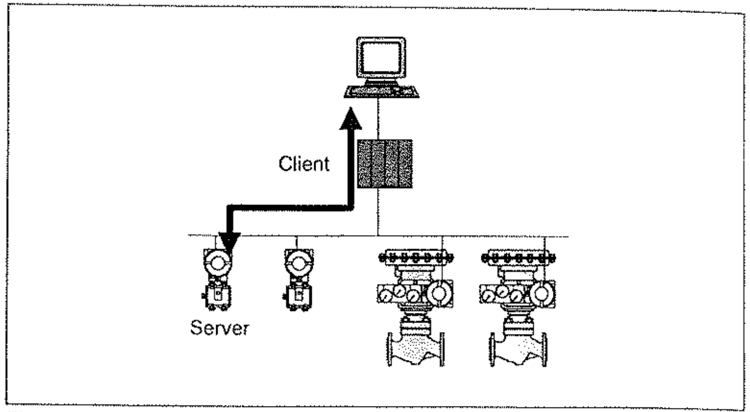


Figure 11-17. Client-server VCR.

has an index. Every element of a parameter has a subindex. The OD maps to the "real" data in the device memory and the data type of the parameter. A user never sees device addresses, VFDs, or indexes since the user interacts with the device solely according to block tags and parameter names.

FOUNDATION Virtual Field Device (VFD)

A virtual field device (VFD) is a logical subdivision of the information that is accessible from a device, that is, smaller virtual devices within the physical device. Every FOUNDATION device consists of at least two VFDs. The first VFD contains System Management (SM) and Network Management (NM) information. The second and any subsequent VFDs are used to access the Function Block Application Process (FBAP), that is, function blocks, resource blocks, and transducer blocks in the device (figure 11-18).

FOUNDATION Communication Services

The FOUNDATION FMS provides a number of services for reading, writing, and otherwise accessing objects. There are seven groups of services targeted for different objects:

VFD support services are used to manage virtual field devices. They do the following:

- Read a device/user status
- Report an unsolicited status
- Read vendor, type, and version
- FMS reject of improper confirmed service

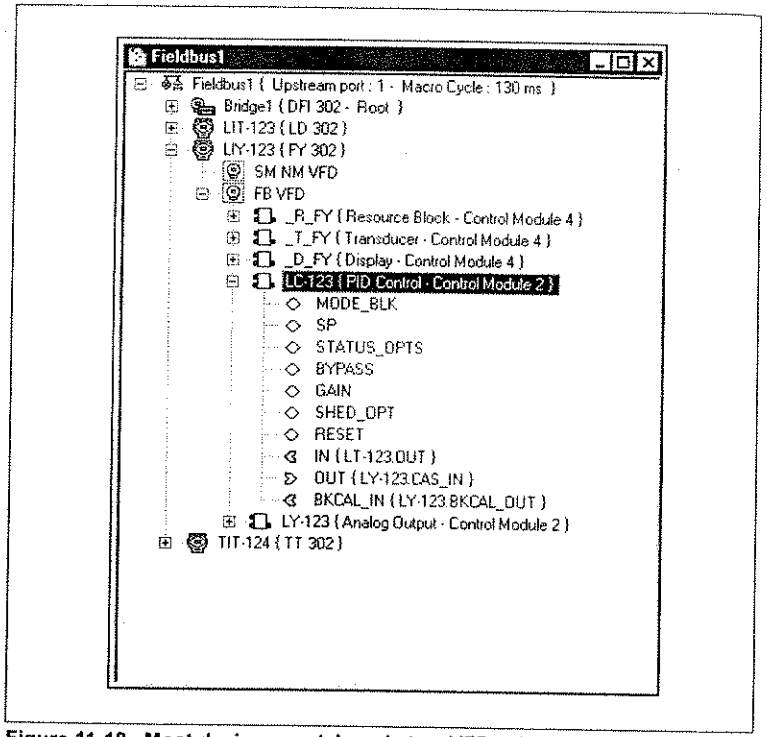


Figure 11-18. Most devices contain only two VFDs: SM/NM and FBAP.

Object dictionary (OD) management services are used to read and write parameters and download configurations to the OD. They do the following:

- Read an object description
- Start an OD Load
- Load an object description
- Stop an OD Load

Context management services are used to manage VCRs. They do the following:

- Establish a connection
- Release a connection

Domain management services are used to download and upload data into a memory area in a device. They do the following:

- Open a download sequence
- Transmit a download data block
- Stop a download sequence
- Open a download sequence
- Transmit a download data block
- Stop a download sequence
- Request a download
- Open an upload sequence
- Transmit an upload data block
- Stop an upload sequence
- Request an upload

Program Invocation (PI) management services are used to execute and manage a program in a device. They do the following:

- Create a program
- Delete a program
- Start a program
- Stop a program
- Resume a program
- Reset a program
- Kill a program

Variable access services are used, for example, to read the view objects associated with blocks. They do the following:

- Read a variable
- Write a variable
- Read a variable with type
- Write a variable with type
- Read a memory location
- Write a memory location
- Publish data
- Report data
- Publish data with type
- Report data with type

- Define a variable list
- Delete a variable list

Event management services are used to report and manage alarms, events, and trends. They do the following:

- Report an event
- Report an event with type
- Acknowledge an event
- Disable/enable an event

PROFIBUS DP Communication Profile

PROFIBUS PA is based on the PROFIBUS DP communications profile. It is not an application layer but is very similar to one and is therefore included in this section.

DP Communication Relationships

As far as the PROFIBUS DP is concerned there are two types of transports:

- Cyclic communication
- Acyclic communication

Cyclic communication is used for data that has to be updated continuously through recurring read and write requests to the slaves. Cyclic communication includes, for example, process inputs and outputs, which are typically scanned by a class 1 master such as a programmable controller.

Acyclic communication is used for data that only has to be communicated infrequently. Acyclic communication includes, for example, the host reading and writing parameters in the field instruments and downloading and entire device configuration, typically from a class 2 master such as a host configuration tool.

Cyclic communication has the highest priority on the network. Usually, the token rotation time is increased during acyclic communication. This changes the loop dynamics and should be taken into account when tuning control loops.

DP Slot and Index

PROFIBUS DP uses a concept by which a device consists of one or more modules. The modules can be physical or logical subdivisions of the device, that is, smaller virtual devices within the physical device. Simple devices like transmitters and positioners using PROFIBUS PA typically only have a single slot. Complex devices have many slots. In PROFIBUS PA, one function block is mapped into one slot. Information in the devices is accessed in the form of modules or blocks, which are addressed by slot number. A slot may at the same time also contain the physical block and transducer blocks. Each parameter in the block is identified by an index. The slot and index maps to the "real" data in the device memory.

DP Communication Services

PROFIBUS DP provides a number of "services", functions, that master and slave devices use for reading, writing, and otherwise accessing objects. One of the communication services is selected based on what has to be done. E.g. for a programmable controller to read the process value from a transmitter it may select the service for a master class 1 to read data from a slave.

Services for acyclic transmission between master class 1 (central controller) and slave:

- Master class 1 reads data from slave
- Master class 1 writes data to slave
- Alarm transmission from slave to master class 1
- Master class 1 acknowledges alarm to slave
- Status transmission from slave to master class 1

Services for acyclic transmission between master class 2 (engineering tool) and slave:

- Establish connection for data transport
- Terminate connection for data transport
- Master class 2 reads data from slave
- Master class 2 writes data to slave
- Master class 2 writes and reads data to/from slave over connection

Services for cyclic transmission:

- Data exchange for function block inputs and outputs
- Check consistency between master and slave exchange configuration
- Select parameters for cyclic exchange
- Master class 2 reads data from slave
- Master class 2 writes data to slave
- Master class 1 reads data from slave
- Master class 1 writes data to slave
- Indicate diagnostics
- Set slave address

Only one master is allowed to write outputs. It is therefore a good idea to verify that the redundancy scheme employed in the system correctly switches over in the event of primary failure.

System Management and Network Management

HART and PROFIBUS PA do not use system and network management. System and network management in the devices and the host work together to provide functions like automatic address assignment and clock synchronization. Clock synchronization is important because time is used to time-stamp alarms and schedule function block execution and communication.

FOUNDATION Fieldbus H1

The system management in a device is responsible for function block and communication scheduling, time synchronization, and automatic address assignment, among other functions. The Network Management Information Base (NMIB) and System Management Information Base (SMIB) are accessed through the first VFD in the device. The NMIB contains the LAS communication schedule, VCRs, communication statistics, and the like. The SMIB contains the device's tag, address, function block schedule, and so on.

Function Block Schedule

The schedule for function block execution and link communication is explained in chapter 4, "Configuration." The schedule is stored in the SM/NM VFD in the device. SM coordinates the scheduling of function block execution and the communication of function block links in order to eliminate jitter. This is because a value is published when it is ready without having to delay for an inexact period of time. When the point is reached in the schedule where a function block link should be published the LAS will transmit a Compel Data (CD) message to the concerned device, which subsequently broadcasts the link.

Automatic Address Assignment

Every FOUNDATION device has a 32-byte unique identifier, which is a hardware address that consists of a 6-byte manufacturer code, 4-byte device type code, and a serial number. These uniquely distinguish the device from all others in the world. The manufacturer code is universally administered by the Fieldbus Foundation, which eliminates the potential for duplication. The manufacturer assigns the device type code and sequential number.

The LAS periodically polls network addresses to detect new devices. LAS identifies new devices and assigns them an operating address in case the original address was in the default range or a duplicate address is being used by a device already on the network. Network address conflicts are automatically resolved because the LAS uses the unique identifier hardware address to distinguish between the devices contending for the same network address.

Clock Synchronization

The system management in the LAS periodically publishes the time for the devices on the network to synchronize their internal clocks. This time is used to ensure that the schedule in each device is synchronized with all the others, which ensures that the blocks are executed and the links published at exactly the right moment. The time is also used to time-stamp alarms, events, trends, and the like.

Should the primary time publisher fail a secondary will take over and so on.

User Layer

The user layer is where the real functionality of the device or software takes place. This is where transmitters measure, positioners actuate, and hosts interface to the user. It is at the user-layer level that the data formats and semantics are defined that allow devices to understand and act intelligently on data. Thus, real interoperability can be achieved. Both FOUNDATION Fieldbus and PROFIBUS

PA have an object-oriented user layer that is based on blocks. Object techniques are used extensively. For example, the concept of encapsulating is seen in encapsulation of parameters and functionality into blocks. The concept of inheritance is seen for the universal parameters inherited in every block and the common behavior and parameters within the four classes of blocks. The concept of instantiation is used for creation of multiple uniquely tagged blocks. For example, within the PID block all the functionality and parameters related to the control are contained in the block rather than scattered throughout the device memory. The blocks are described in detail in chapter 4, "Configuration."

FOUNDATION Fieldbus

The FOUNDATION user layer is based on blocks distributed among the devices on the network. Within a device the blocks are contained in a Function Block Application Process (FBAP) within a VFD.

FOUNDATION Block Objects

The FBAP is divided into a device application process (DAP) and control application process (CAP). The DAP contains the resource block and the transducer blocks for the device configuration, and the CAP contains the function blocks that make up the control strategy (figure 11-19).

The control strategy that is built using the function block programming language usually spans several devices. The user selects the type of blocks that are required for the application and names them using unique tags. The number and types of blocks that can be instantiated in the device may be fixed in memory or blocks may be dynamically instantiable, meaning that the quantity of blocks can be freely selected from a library of block types.

Resource block is responsible for the overall management of the device, such as running it or placing it out of service, as well as forcing the output. It also contains general identification information and overall device diagnostics.

Transducer blocks are responsible for interfacing the sensor, actuator, and display hardware to the function blocks. This is where calibration is done. They also contain information about the last calibration and I/O diagnostics.

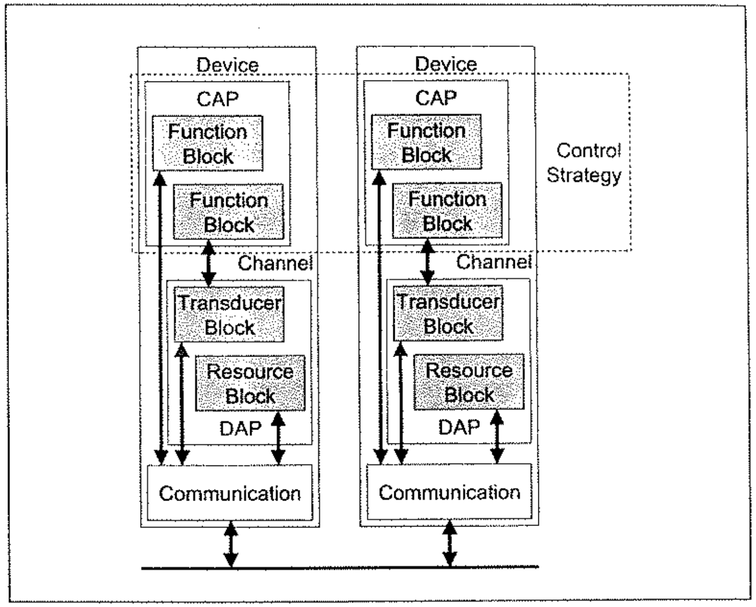


Figure 11-19. Device configured in resource and transducer blocks, control strategy in function blocks.

Function blocks are responsible for executing input, output, control, and calculation functions. Function blocks can be linked together to form a complete control strategy.

FOUNDATION Link Objects

Function blocks have outputs that can be linked to the input of other blocks, thus propagating information through the control strategy. The definitions of links between blocks in the same or different devices are contained in link objects.

FOUNDATION Trend Objects

Trend objects sample block parameters for the sake of historical trending. Multiple samples are collected and communicated in a single communication to the host.

FOUNDATION Alert Objects

Alert objects report alarms and events detected in the function blocks to the host. The host must confirm receipt of the alert. If it does not, the alert is retransmitted.

FOUNDATION View Objects

View objects are predefined subsets of block parameters that allow the parameters within the same block to be read using fewer communications. There are four different views:

VIEW_1 is a subset of common dynamic parameters in the block that are suitable for operation from a faceplate.

VIEW_2 is a subset of common static parameters in the block that are suitable for operation from a faceplate.

VIEW_3 is a subset of all dynamic parameters in the block.

VIEW_4 is a subset of the static parameters in the block not already included in VIEW_2.

FOUNDATION Multi-Variable Container (MVC)

The Multi-Variable Container (MVC, see chapter 7) can be configured to access a user-defined subset of parameters across various blocks within a device. For example, if a device has sixteen inputs all of these can be accessed with a single communication. Usually, the host configures the MVC automatically depending on what values from the device are displayed in the operator console.

FOUNDATION Device Description (DD)

Device Description (DD) describes all data in the device. Its practical use is explained in chapter 3, "Installation and Commissioning." The device manufacturer writes DD using the Device Description Language (DDL). It then compiles it and supplies it as two files together with the device. The DD allows the host to interpret the complex data in the devices so it can be displayed in a comprehensible way. The DD describes every object in the device. Indeed, HART also has DD that is very similar to that of the FOUNDATION. DDL has fourteen constructs with associated attributes to completely describe the data and even suggest how it should be presented to the operator. DD is used in conjunction with a capability file. The device manufacturer writes the capabilities file in the Common File Format (CFF). The capabilities file defines the resources that are available in the device. This ensures that the host does not exceed available resources when configuring off line. The DDL constructs of the DD file include fourteen terms, described in the following paragraphs:

Variable describes every piece of information in the device in terms of the following characteristics:

- Input, output or contained
- Dynamic or static
- If related to diagnostics, service, operation, alarm or tuning
- Data type: integer, float, enumerated, ASCII, time, etc.
- Significant digits and decimals
- Minimum/maximum value
- Scaling factor
- Text for enumerated and bit-enumerated variables
- Fixed engineering unit
- Read-only
- Help text for user
- Label identifying the variable to the user
- Methods before and after editing
- Methods before and after reading
- Methods before and after writing
- Read/write timeout
- When applicable
- Error message to the user

Item Array describes a logical grouping of data in the device:

- Objects that are part of the item array
- Help text for user
- Label identifying the item array to the user

Collection describes a logical grouping of data in the device:

- Objects that are part of the collection
- Help text for user
- Label identifying the collection to the user

Record describes data in the device grouped for communication:

- Objects part of the record

- Help text for user
- Label identifying the record to the user
- Error message to the user

Array describes data in the device grouped for communication:

- Type
- Number of elements
- Help text for user
- Label identifying the array to the user
- Error message to the user

Variable List describes data in the device grouped for communication:

- Objects part of the variable list
- Help text for user
- Label identifying the variable list to the user

Block describes every block type in the device:

- Type and execution time, etc.
- Label identifying the block to the user
- Parameters part of the block
- Help text for user
- Variable lists part of the block
- Objects associated with the block

Menu describes how the data is organized when displayed to the user:

- Label identifying the menu to the user
- Objects part of the menu

Edit Display describes how the data is displayed to the user when editing (for example, when the user edits the range Edit Display allows the range limits to also be displayed):

- Object to be edited
- Label identifying the edit display to the user
- Objects to be displayed to assist the user
- Methods before and after editing

Method describes procedures for the configuration tool and the user when performing functions in the device (for example, when calibrating an output, the output may first have to be set to one extreme, the user prompted for a calibrator reading, the adjustments made, and device operation finally returned to normal, etc.):

- Input or output
- If related to diagnostics, service, operation, alarm, or tuning
- ANSI C language to execute
- Help text for user
- Label identifying the method to the user
- When applicable

Relation describes how data that belongs together needs to be synchronized. For example, when the engineering unit is changed the range limits may need to be refreshed because they change, even though they are static parameters. Changing the sensor type may also affect the range limits.

- Need to be updated when one in a group is changed etc.
- Need to be updated when the unit is changed
- Need to be written at the same time

Program describes how a program in the device can be operated:

- Data to be passed to the program
- Error message to the user

Domain describes how memory in the device can be handled:

- Read-only
- Error message to the user

Response Code describes error messages to the user.

Additionally, some DD identification information is included. The DDL language also has a number features for controlling the DDL compiler.

PROFIBUS PA Application Profile

The PROFIBUS PA user layer is based on blocks distributed among the devices on the network. Within a device, the blocks are contained in a virtual module that is accessed by slot number.

PA Block Objects

The configuration is stored in the resource block, transducer blocks, and function blocks.

Physical block is responsible for the overall management of the device, such as running it or placing it out of service, as well as forcing the output. It also contains general identification information and overall device diagnostics.

Transducer blocks are responsible for interfacing sensor, actuator, and display hardware to the function blocks. This is where calibration is performed. It also contains information about the last calibration and I/O diagnostics.

Function blocks are responsible for executing the input and output part of the control strategy.

PA Link Objects

Function blocks have outputs that can be linked to the input of other blocks, thus propagating information through the control strategy. The definition of the links between blocks is contained in link objects. The PROFIBUS PA, version 3, only supports links within the device.

PA View Objects

View objects are predefined subsets of block parameters that allow the parameters within the same block to be read using fewer communications. There are four different views:

VIEW_1 is a subset of common dynamic parameters in the block that are suitable for operation from a faceplate.

VIEW_2 is a subset of common static parameters in the block that are suitable for operation from a faceplate.

VIEW_3 is a subset of all dynamic parameters in the block.

VIEW_4 is a subset of the static parameters in the block that are not already included in VIEW_2.

Fieldbus on Ethernet and IP

There is no HART protocol for the host-level network. Therefore, HART will have to be used with some other network technology. FOUNDATION Fieldbus and PROFIBUS both have host-level networking solutions built on the Ethernet platform. Ethernet corresponds to the physical and data link layer in the OSI model. The Internet Protocol (IP) corresponds to the network layer. The User Datagram Protocol (UDP) and Transmission Control Protocols (TCP) correspond to the transport layer (figure 11-20).

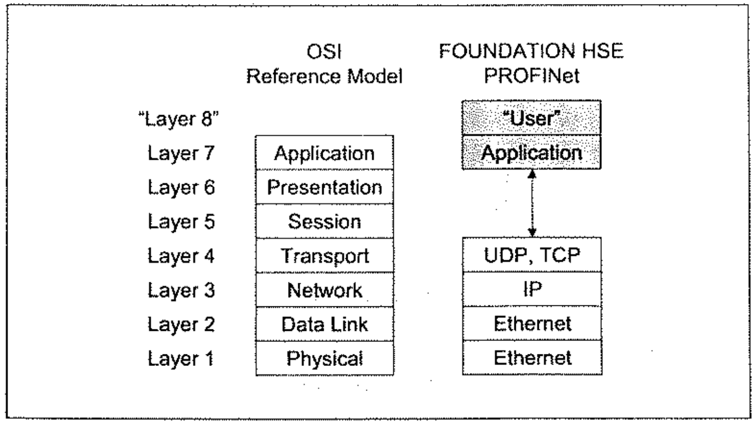


Figure 11-20. Fieldbus technologies on Ethernet and IP.

These four layers are not sufficient to provide interoperability. Application layer and user layer are also required to achieve a completely interoperable protocol because the formats and semantics of the data are defined in these higher layers. FOUNDATION HSE Fieldbus and PROFInet put the existing FOUNDATION and PROFIBUS technologies on top of the Ethernet and IP platforms. PROFInet on top of TCP uses the Microsoft DCOM (Distributed component object model) technology for communication.

Ethernet, IP, UDP, and TCP are discussed in many books. We therefore won't detail these technologies here.

FOUNDATION HSE Fieldbus

FOUNDATION HSE Fieldbus supports both the TCP and UDP transport layer protocols, but UDP is the default means. The same functionality provided by the FOUNDATION application and user layers is also provided over the Ethernet.

HSE Application Layer

In HSE, the application layer is called the Field Device Access (FDA) agent. It provides the Virtual Communication Relationship (VCR) functionality that is the equivalent of what FAS does for H1, that is, client/server, publisher/subscriber, and report distribution. In other words, over Ethernet the blocks in the VFDs are accessed through the FDA agent. The Ethernet data link layer does not support scheduling, hence publisher/subscriber data is transmitted on an "as soon as possible" basis.

HSE Network Management

HSE network management is based on standard Internet technologies and uses the standard Simple Network Management Protocol (SNMP) to access a standard Management Information Base (MIB-II) with some extensions for HSE. The HSE devices also support the Dynamic Host Configuration Protocol (DHCP) for assigning IP addresses automatically. The Simple Network Time Protocol (SNTP) is used to synchronize clocks. The time received over SNTP is used to set the clock in the HSE device, and a linking device also uses this time to synchronize the clocks in the devices on the H1 networks that are connected to it. Clock synchronization makes possible scheduled function block execution in HSE devices.

SNMP, DHCP, and SNTP are discussed in numerous books so we will not detail them here.

HSE LAN Redundancy Entity (LRE)

Redundancy is used to improve the availability of the host-level network by providing not only redundant media but also redundant communication ports and devices. In the device, the LAN Redundancy Entity (LRE) is responsible for redundancy. Redundant devices have multiple physical network connections but only use one for the normal communications at any one time. At the same time, all other connections take part in diagnostics communication to verify the integrity of the system. Should the network, port, or one part of the redundant device fail, the normal communication will take another path. There is no central redundancy manager for the network. Instead, the LRE takes care of the switchover from primary to secondary based on detected faults. The switchover is transparent to the VFD and FBAP, so no special configuration of function blocks or otherwise is required. When the failed primary device is once again OK it returns on the network as a secondary.

The LRE in the HSE devices periodically transmits to the other devices diagnostic messages about which devices and ports it can see on the network. Each device gathers this information from the other devices in an internal network status table and uses it to determine which path it shall use to communicate with the other devices.

EXERCISES

11.1 Which has a sinusoidal signal, HART or IEC?

11.2 Which signal is self-clocking, HART or IEC?

11.3 What are the three classes of HART commands?

11.4 What are the two types of communication in PROFIBUS PA (DPv1)?

11.5 What entity is responsible for automatically assigning addresses?

11.6 Which file(s) defines how many blocks of a given block type a FOUNDATION device supports?

11.7 Which type of block is responsible for the overall operation of a PROFIBUS PA device?

11.8 Which transport protocol does HSE use?

11.9 Which entity is responsible for HSE redundancy?

BIBLIOGRAPHY

1. HART Communication Foundation, *HART Field Communications Protocol: An Introduction for Users and Manufacturers*, July 1997.

2. ISA-50.02, Part 2-1992 *Fieldbus Standard for Use in Industrial Control Systems. Part 2: Physical Layer Specification and Service Definition*. Research Triangle Park, NC: ISA, ISBN 1-55617-317-2, 1992.

3. Technical Description, Order Number 4.002, PROFIBUS Nutzenorganisation e.V., September 1999.

4. *Technical Overview*, FD-043 Revision 2.0, Fieldbus Foundation, 1998.

Appendix A
Solutions to Exercises

Chapter 1

1.1 No. For example, telephone and television systems are networks that, at least for now, are still very analog.

1.2 Yes.

1.3 Yes.

1.4 Yes, it requires fewer transactions. Master/slave needs to make one read transaction and then one write transaction for each recipient, whereas publisher/subscriber only needs a single direct transaction from originator to all recipients.

1.5 Factory automation.

1.6 No. It is just a physical and data link layer, but nevertheless an essential platform for many protocols.

1.7 Yes. Any fault affects a smaller part of the plant.

Chapter 2

2.1. No.

2.2. The device manufacturer.

2.3. Transducer block.

2.4. Yes. For example, if the operator is unaware that a positioner has failed, this could be a hazard. Diagnostics tell the operator that the loop is not under control.

2.5. Floating-point values, to follow the IEEE 754 standard.

2.6. No, not when communicating digitally for direct measurements. If you use the 4-20 mA associated with HART you must set the range. The same is true if you use fieldbus for inferred measurements.

2.7. Tag.

2.8. Yes. For example, more precise diagnostics will indicate the severity level of the fault, thus helping plants avoid shutdowns for noncritical problems.

2.9. Yes. But the particular technology and the device should be certified.

Chapter 3

3.1. 230 ohm.

3.2. No.

3.3. 0 (zero).

3.4. A network consists of one or more segments. A link is the same as a network, and it is also another name for a linking device.

3.5. No. It also acts as a shunt.

3.6. No.

3.7. No.

3.8. Yes. For thirteen spurs the table states 90 meters, but for twelve spurs it states 120 meters. However, the rules are not strict. Therefore, it is reasonable to assume that for some of the devices the cable can safely be somewhere between 90 and 120 meters.

3.9. No.

3.10 A linking device buffers messages to help the system handle the speed difference between the field-level and host-level networks. A coupler forces the host-level network to run at a speed close to that of the field-level network. A link is the same as a linking device; it may also be another name for a network.

3.11. Four devices. Although 60 divided by 12 equals 5, 5 is really at the very limit. Four devices would leave you some margin and allow you to use longer cable.

3.12. Yes.

3.13. Trapezoidal.

3.14. Yes, by using barriers with safe side repeaters.

3.15. Two. But a network can have many segments.

3.16. Switched.

3.17. There is no limit.

3.18. Router.

3.19 Yes. If one cable is broken there is a second path.

Chapter 4

4.1. A bridge.

4.2. 700, obtained as follows: 500 - -200.

4.3. The LRV becomes 50. To maintain a span of 700 the URV becomes 750.

4.4. Device configuration is performed in the resource (physical) and transducer blocks. Function blocks are used to configure control strategy, but in the case of PROFIBUS PA they may also be considered part of the device configuration.

4.5. Nonvolatile.

4.6. Simulation active is set in the block error parameter in the resource block.

4.7. MODE_BLK.Target for FOUNDATION devices and TARGET_MODE for PROFIBUS PA devices.

4.8. Quality, limit, and substatus.

4.9. No.

4.10. In the transducer error (XD_ERROR) parameter.

4.11. In the primary value (PRIMARY_VALUE) parameter.

4.12. To the remote cascade input (RCAS_IN) parameter.

4.13. Dynamically instantiable blocks.

4.14. In the positioner.

4.15. An alarm both occurs and clears, whereas an event only occurs.

4.16. 4,9-103,0 kPa.

4.17. Good(Cascade) and Good(noncascade).

4.18 No. It requires power in order to function. Therefore, the final control element must deenergize safely.

4.19 The status options (STATUS_OPTS) parameter.

4.20 Local override (LO) is the fail-safe mode in which the setpoint (SP) parameter is written.

4.21 The option parameters can only be written in out-of-service (OOS) mode.

4.22. Initialization manual (IMan) is the mode in which the output is back-calculated in order to get a bumpless transfer.

4.23. Yes.

4.24. Enable option Target to manual for Bad IN in the status option (STATUS_OPTS) parameter.

4.25. Set the linearization type (L_TYPE) parameter as Direct.

4.26. The OUT_*_LIM limits are more constrictive than OUT_SCALE.

4.27. The output (OUT).

4.28. In the PID block control option (CONTROL_OPTS) parameter.

4.29. No. The physical output is not changed by the simulation. It is a readback signal that is simulated.

4.30. Availability. The default option for fault state in the I/O-option (IO_OPTS) parameter is fault state to value.

Chapter 5

5.1. Yes. However, there must be a power supply impedance between the regular DC power supply and the IEC 61158-2 network. The power supply impedance must be able to handle any noise or instability in the DC power supply.

5.2. No.

5.3. Yes. However, the OPC server should be safety approved. Most likely, it is restricted to read-only operation.

Chapter 6

6.1. 9 VDC according to the standard, although some devices require more.

6.2. Most likely, the mode of the resource block is out-of-service.

6.3. 120 m. The wire from the interface to the terminator is a spur like any other. However, most of the time the terminator, power supply impedance, and interface are very close together.

6.4. Typically, eight, but to enjoy the best performance perhaps a bit less, which can be accomplished using some devices for monitoring and auxiliary functions.

Chapter 7

7.1. Yes. If the process value appears unreasonable but the status is "Good," then the device must be OK, so it must therefore be a true process problem.

7.2. No. For bit-enumerated parameters multiple options are possible.

7.3. Yes. If several parameters are accessed from the same block then they can be accessed using a View object in a single communication.

7.4. No.

Chapter 8

8.1. No, not if you use devices that have sufficient resources.

8.2. Three: two for Device Description and one Capabilities file.

8.3. No, not in the opinion of this author. A plant can simplify engineering and purchasing by using a single standard requirement for each device type throughout the plant.

8.4. Device and wiring details should all be in the network diagram. Only control functionality should appear in the P&ID.

8.5. No. Provided that the templates are created by using standard function blocks they can be assigned to any device that uses standard function blocks. However, if any devices do not support standard blocks, or templates are made using nonstandard blocks, the blocks cannot be universally applied.

Chapter 9

9.1. In the transducer blocks. Range setting is performed in the function blocks.

9.2. The block error (BLOCK_ERR) parameter.

9.3. A large number of cycles is an indication of an unstable process.

9.4. The standard function block types defined are, but extended function blocks and transducer blocks are not. However, transducer blocks have few parameters to be set and no links, so they are easy to change.

Chapter 10

10.1. No. The goal of availability is to keep a system running, whereas for safety the goal is to shut down.

10.2. Yes. Better diagnostics reduces false trips and allows true faults to be detected and rectified faster.

10.3. Yes. Better diagnostics detects dangerous faults.

10.4. No, not the way it is implemented in control systems. Redundancy improves availability.

10.5. No, not if the loop is configured to shut down safely.

10.6. The same value used in a dangerous loop can shut it down, while a nondangerous loop is allowed to continue.

10.7 No. It improves availability.

Chapter 11

11.1. HART (Bell 202).

11.2. IEC (Manchester coding).

11.3. Universal, common practice, and device specific.

11.4. Cyclic and acyclic.

11.5. System management.

11.6. The Capabilities file.

11.7. The physical block.

11.8. UDP.

11.9. LRE.

Appendix B

Parameters Index

Index