Technical Information

TI 38K02A01-01E

FOUNDATION™ fieldbus Book - A Tutorial

Yokogawa Electric Corporation
2-9-32, Nakacho, Musashino-shi, Tokyo, 180-8750 Japan
Tel.: 81-422-52-5634 Fax.: 81-422-52-9802

TI 38K02A01-01E

3rd Edition Apr. 2012(YK)

Blank Page

Introduction

This document is authored by Yokogawa Electric Corporation and its group for those who wish to learn more about technologies supporting FOUNDATION fieldbus™ and its function blocks. It is to encourage adoption of FOUNDATION fieldbus.

There is no such intention to override any technical content of FOUNDATION fieldbus. If this textbook contains any explanation that conflicts with documents published by the Fieldbus Foundation, their documents are the master.

Yokogawa hopes that this document encourages more people to apply the FOUNDATION fieldbus in their industrial automation projects.

■ Document structure and target readership

This textbook consists of following three chapters:

Chapter 1 FOUNDATION fieldbus Overview

Chapter 2 Fieldbus Communication Technologies

Chapter 3 Fieldbus Applications

Chapter 1 provides a brief overview of FOUNDATION fieldbus.

Chapter 2 describes FOUNDATION fieldbus communication technologies. Start-up engineers and field engineers must understand the basics of the communication technologies explained in this section.

Chapter 3 is about applications running on the fieldbus. They are essential for control and measurement of the digital networks. Those engineers who are involved in the instrumentation with digital communication need those knowledge for project implementations.

Trademarks

- "FOUNDATION" in "FOUNDATION fieldbus" is a trademark of the Fieldbus Foundation.
- CENTUM, YTA and YVP are registered trademarks of Yokogawa Electric Corporation.
- EJA is a trademark of Yokogawa Electric Corporation.
- Other product and company names may be registered trademarks of their respective companies (the ™ or ® mark is not displayed).

IMPORTANT

DISCLAIMER OF WARRANTIES

This document is provided on an "as is" basis and may be subject to future additions, modifications, or corrections depending on the results of field testing. Yokogawa Electric Corporation hereby disclaims all warranties of any kind, expressed or implied, including any warranty of merchantability or fitness for a particular purpose, for this document. In no event will Yokogawa Electric Corporation be responsible for any loss or damage arising out of or resulting from any defect, error or omission in this document or from anyone's use of or reliance on this document.

This document refers to the open communication fieldbus specifications developed by the Fieldbus Foundation, an international organization that major control and instrumentation vendors and users around the world participate and operate, based on the International Electrotechnical Commission (IEC) standards.

FOUNDATION Fieldbus Book
- A Tutorial

TI 38K02A01-01E 3rd Edition

CONTENTS

1. FOUNDATION fieldbus Overview 1-1
1.1 What is fieldbus? 1-1
1.2 Benefits of fieldbus 1-3
1.3 FOUNDATION fieldbus 1-5
2. Fieldbus Communication technologies 2-1
2.1 Communication models 2-2
2.1.1 OSI reference model 2-2
2.1.2 Protocol data unit 2-4
2.1.3 Communication through virtual communication relationship 2-5
2.2 Physical layer 2-6
2.2.1 Physical layer of 31.25 kbps 2-6
2.2.2 Signaling Method 2-7
2.2.3 Wiring rules 2-8
2.2.4 Intrinsic safety (IS) consideration 2-10
2.3 Data link layer 2-11
2.3.1 Medium access control 2-11
2.3.2 Addresses 2-12
2.3.3 Link active scheduler 2-14
2.3.4 Scheduled communication 2-15
2.3.5 Unscheduled Communication 2-16
2.3.6 Link Maintenance 2-17
2.3.7 Data Link PDUs 2-17
2.4 Application layer 2-18
2.4.1 Fieldbus access sub-layer 2-18
2.4.2 Fieldbus message specification 2-22
2.5 System management protocol 2-26
2.5.1 Tag and address assignment 2-26
2.5.2 Tag location 2-26
2.5.3 Application time synchronization 2-26

3. Fieldbus Applications 3-1
3.1 Virtual field devices 3-1
3.1.1 VFDs in a fieldbus device 3-1
3.2 FOUNDATION fieldbus function block 3-2
3.2.1 What is an FF function block? 3-2
3.2.2 Function block linking and scheduling 3-6
3.2.3 Parameters 3-7
3.2.4 Important parameters 3-9
3.2.5 View objects 3-9
3.3 Important function blocks 3-10
3.3.1 AI block 3-10
3.3.2 AO block 3-15
3.3.3 PID block 3-18
3.3.4 Resource block and transducer block 3-21
3.3.5 Unit Codes 3-23
3.4 System Management 3-25
3.4.1 Device Management 3-25
3.4.2 FF function block management 3-26
3.4.3 Application time management 3-26
3.5 Device support files 3-27
3.5.1 Device description 3-27
3.5.2 Capabilities file 3-30

1. FOUNDATION fieldbus Overview

This section describes an overview of fieldbus before going into the details of the technologies. For more information about the fieldbus, other documents are also available from Yokogawa and in the market.

FOUNDATION fieldbus is also referred to as "the fieldbus" in this textbook.

1.1 What is fieldbus?

The Fieldbus Foundation defines fieldbus as "a digital, two-way, multi-drop communication link among intelligent measurement and control devices." It is one of several local area networks dedicated for industrial automation.

In the digital information era, industries could not survive without information technologies (IT) and networks in the 21st century. Digital communication supports all the economical and social activities by its latest and powerful technologies from production lines to an enterprise level. The fieldbus is a part of digital communication technologies and closely connected with others. Fieldbus is at the lowest level in the hierarchy and exchange information with higher-level data bases.

Standard signals have been defined for the purpose of transmitting control and measuring data in between a central control room and in the field. Standardization of pneumatic and 4 – 20 mA electric signals brought benefits in interoperability and easy maintenance in the early days when industrial automation was introduced.

The smart (hybrid) communication introduced in the mid 1980's, which dawned an era of digital communication, but it had limitations such as proprietary protocols, slow transmission speed, and different data formats.

The idea of developing a common digital communication was proposed to solve these issues. A "standardized" digital communication for industrial automation has rapidly changed the production control systems. Figure 1.1 shows the positioning of the fieldbus in an industrial automation system.

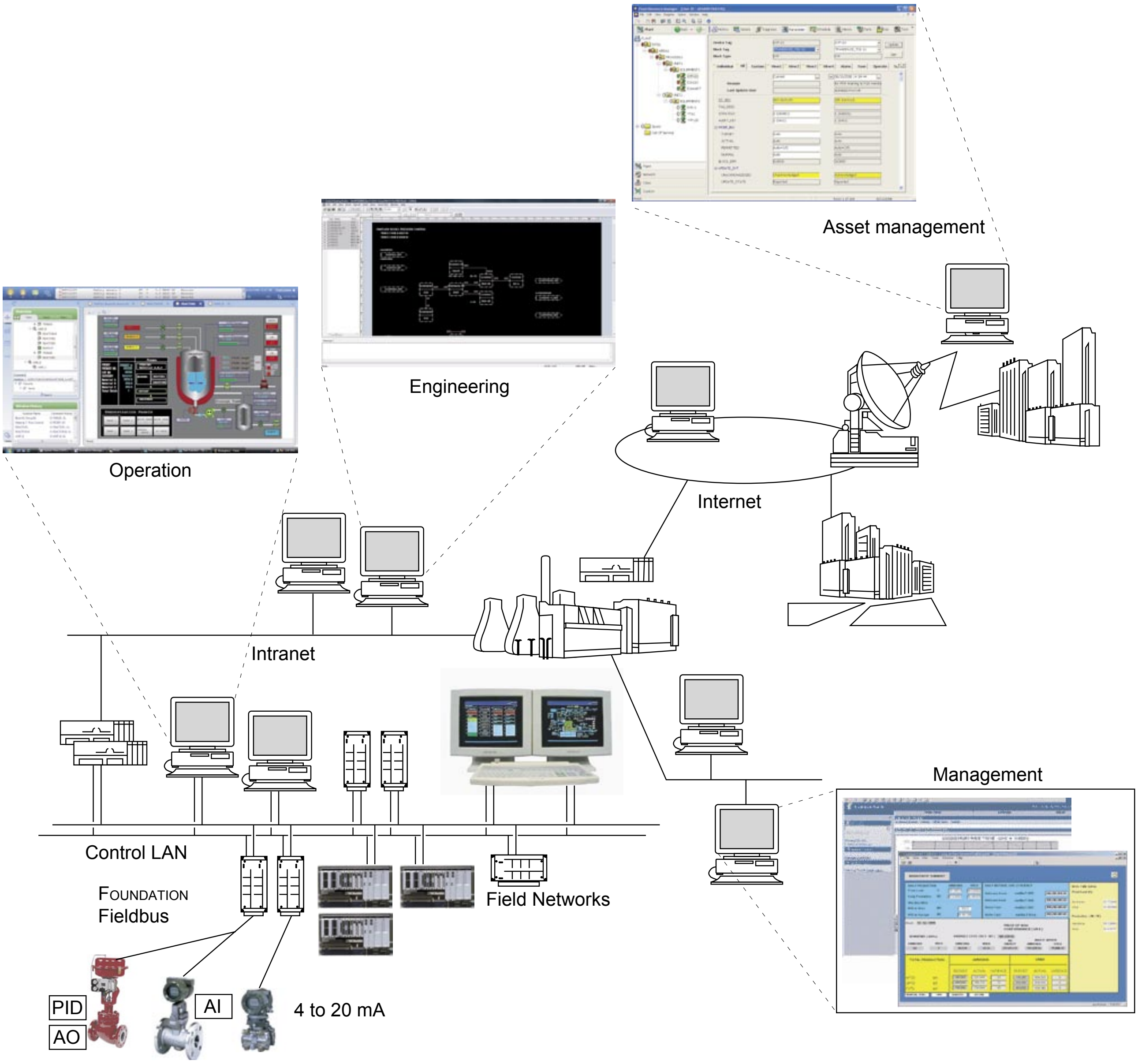


Figure 1.1 Fieldbus positioning

1.2 Benefits of fieldbus

FOUNDATION fieldbus is expected to reduce the plant's total cost of ownership (TCO) by reducing the life-cycle cost of production lines. Figure 1.2 shows the difference between analog transmission and fieldbus communication systems.

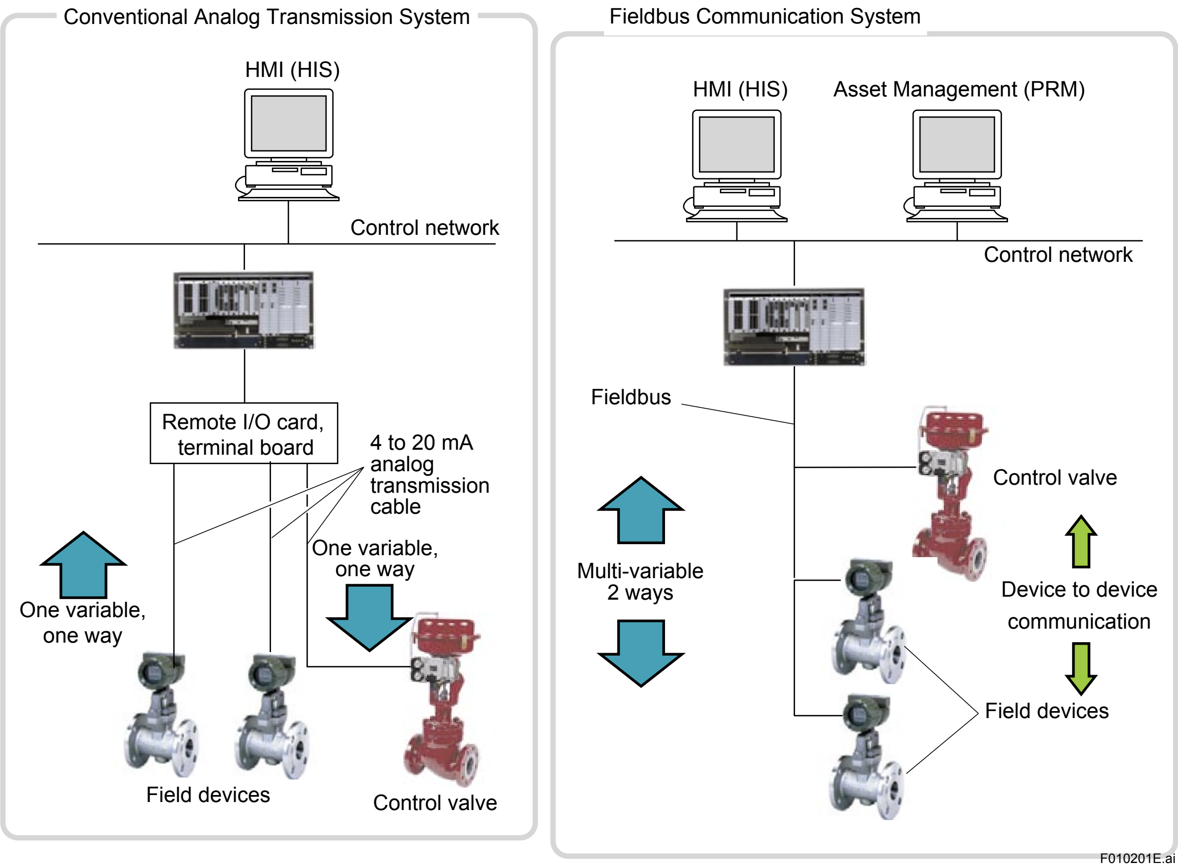


Figure 1.2 Differnce between analog transmission and fieldbus communication systems

■ Benefits in planning

The fieldbus integrates plant equipment into a single plant automation system through its digital communication network. It enables users to enhance information productivity and realize a small control room and a smaller cabinet by connecting devices from various suppliers without customized software.

■ Installation benefits

The fieldbus reduces installation and material cost by replacing the traditional one-to-one wiring scheme with networking or multi-drop connection, while intelligent field instruments shortens commissioning and saves plant startup cost.

Operational benefits

An integrated human-machine interface (HMI) is provided for the plant operation. The fieldbus function blocks enables control functions to be installed on each field device, which enables to transfer the control function to the field.

Benefits in maintenance

The fieldbus enables various notifications related to the self-diagnostics, calibration, and environmental conditions of field instruments without interrupting the plant control. By adopting intelligent field devices equipped with self-diagnostic function, the fieldbus saves a large part of field work by remote access to the device condition information and enable condition-based or proactive maintenance. When used with the asset management software, the maintenance cost can also be minimized.

Benefits in system modification

Functionality of field instruments keeps enhancing endlessly. The fieldbus devices have become the de facto standard and off-the-shelf instruments, which help extend the plant's life cycle cost effectively and without difficultly. The new field devices bring users the benefits of the latest technology. Upgrading of the device firmware can be done online via fieldbus, which contributes in reducing the upgrading cost.

1.3 FOUNDATION fieldbus

The FOUNDATION fieldbus is not a tangible product but a technology available to users to enjoy benefits brought by the fieldbus. The following two conditions must be met to materialize all the benefits:

- Many vendors provide the fieldbus instruments.
- All those devices are interoperable.

The Fieldbus Foundation was established in 1994 to achieve these goals. Its major activities are to:

- Promote a single international fieldbus to both users and vendors,
- Deliver FOUNDATION fieldbus specification,
- Provide technologies for the fieldbus implementation including technical training, and
- Install an infrastructure to achieve interoperability.

The FOUNDATION fieldbus is a subset of IEC standards of IEC61158 and ISA S50.02. The Fieldbus Foundation and its members adopt the FOUNDATION fieldbus as the technology to realize the aforesaid benefits to the users.

2. Fieldbus Communication technologies

In this chapter, the fundamental communication technology that supports the fieldbus function blocks and other applications are explained. Although communication technology, except wiring, is not tangible to users, it is important for users to have precise knowledge in fundamental communication mechanism which helps users understand how the fieldbus function blocks functions in the plant. For those who already have sufficient knowledge about the fieldbus communications or who wish to learn about the fieldbus function blocks quickly, skip this chapter and go to Chapter 3 of this textbook.

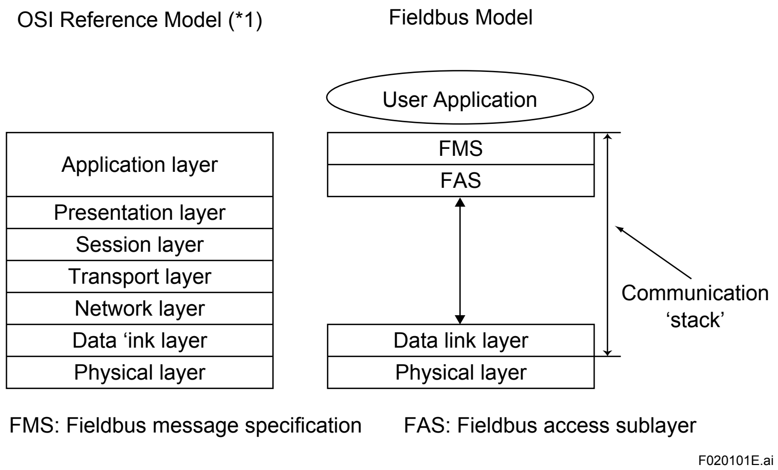
Here in this chapter, how the fieldbus operates and supports users' applications is explained, showing how the fieldbus is cautiously designed to suite for industrial automation applications.

The technology of the FOUNDATION fieldbus described here is based on the specification of the Fieldbus Foundation and a subset of the IEC international standards.

2.1 Communication models

2.1.1 OSI reference model

Communication specifications are often explained by referring to the open system interconnect (OSI) layered model. The communication parts of the FOUNDATION fieldbus is specified according to the simplified OSI model consisting of three (3) layers: physical layer (PHL), data link layer (DLL), and application layer (APL). See Figure 2.1 for OSI reference model and the fieldbus model. From layers 2 to 7 are occupied mostly by software and it is usually referred to as the "communication stack."



*1: The user application is not defined in OSI model

Figure 2.1 OSI Reference Model and Fieldbus Model

The Fieldbus Foundation has defined the Fieldbus Model which includes several user applications which is not included in the OSI Reference Model.

The FOUNDATION fieldbus' application layer consists of two sub-layers: fieldbus access sub-layer (FAS) and fieldbus message specification (FMS). The FAS is the "glue" to map FMS services to the data link layer. Figure 2.2 shows the FOUNDATION fieldbus' architecture.

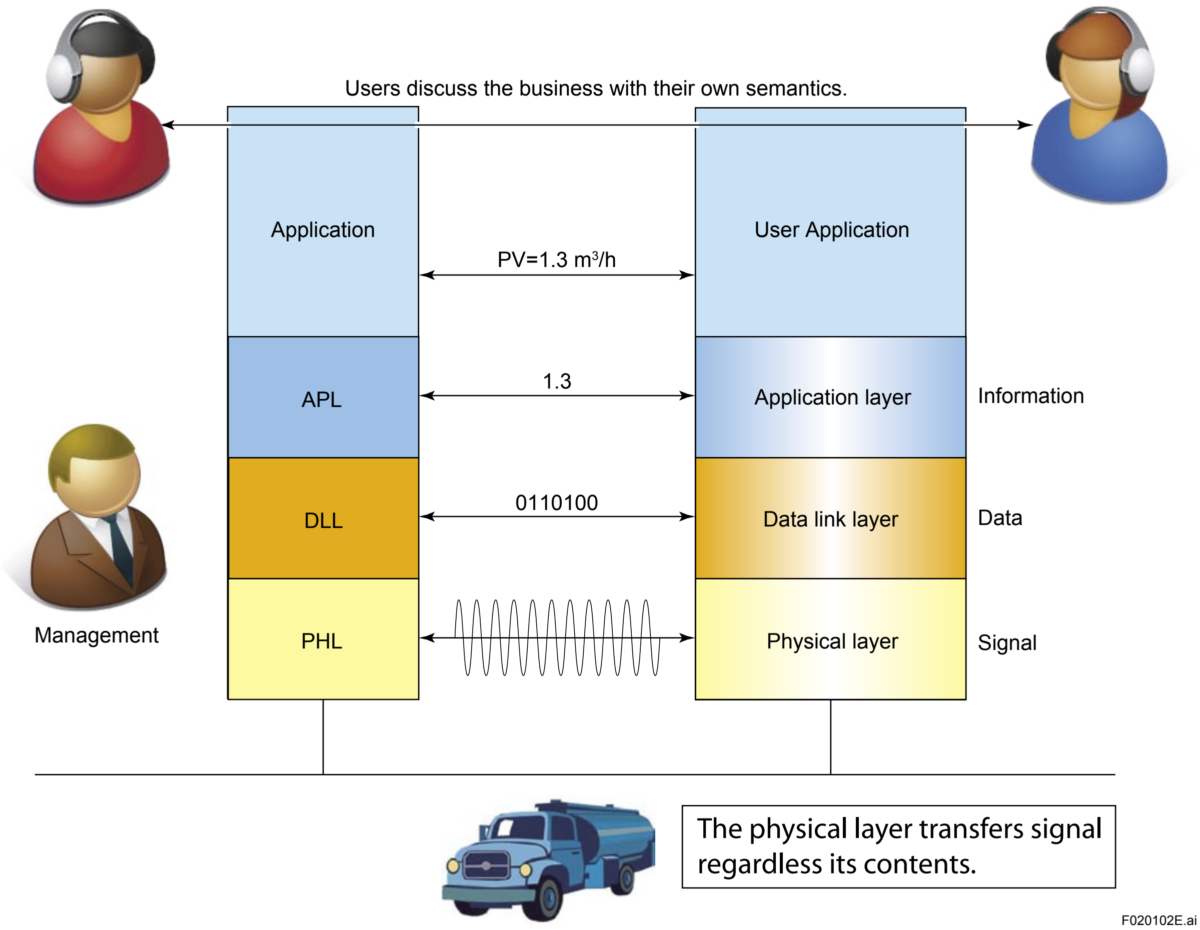


Figure 2.2 FOUNDATION fieldbus architecture

2.1.2 Protocol data unit

Figure 2.3 shows how a user data is transferred through the FOUNDATION fieldbus. Each layer appends layer-control information called protocol control information (PCI). The lower the layer goes, the more information is appended.

A data unit exchanged in between the same layers is called “protocol data unit (PDU).” A PDU may contain an optional data called “service data unit (SDU)” which is passed to and from a higher layer.

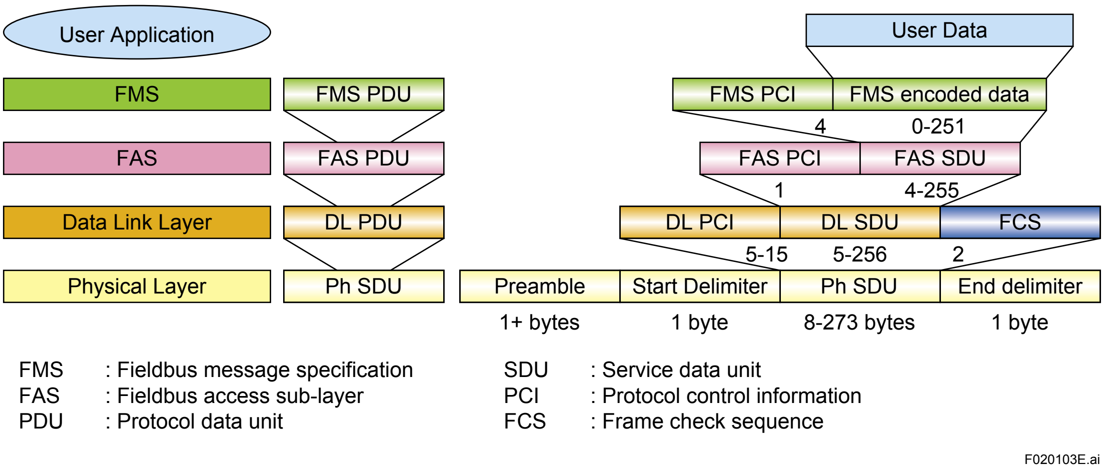
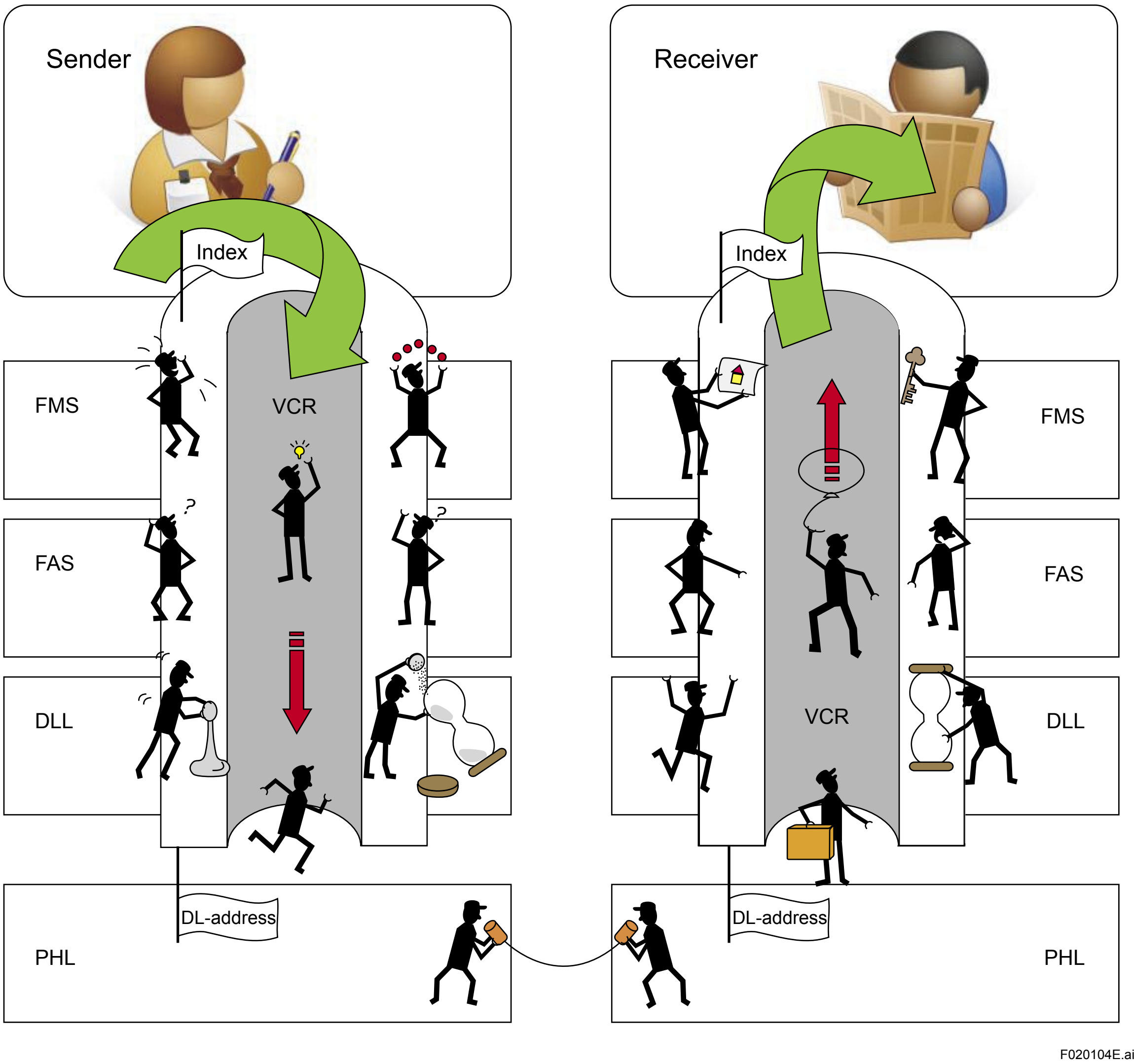


Figure 2.3 User data transmission

2.1.3 Communication through virtual communication relationship

Messages are exchanged among applications resided in the FOUNDATION fieldbus. When a message is transmitted, it goes down through a channel called virtual communication relationship (VCR) to add protocol control information (PCI) before it reaches to the wire. The recipient of the message delivers it to the receiving application through the VCR. PCI is appended or removed when a message passes through VCR allowing each layers to perform its specific functionality. See Figure 2.4.



A fieldbus device has many VCRs which enables communicating with various devices and applications simultaneously. It is made possible that the VCR guarantees transmitting the message to the correct counterpart without risks of losing information. The VCR is identified by an application with "VCR index," a device-local identifier specified in the application layer. It is also identified from other devices with DL-address specified in the data link layer. A single VCR contains a queue (first-in, first-out memory) or a buffer (data storage memory) to save messages. The network configuration is responsible of providing correct index and DL-address together with other operating information to VCRs through network management.

2.2 Physical layer

Physical layer is a mechanism to transmit and receive electric or optic signals to/from medium so that data consisting of ones and zeros is transmitted from one node to the others. The physical layer's specifications are concerned with wires, signals, waveforms, voltages, and all others related to electricity and optics. Among various devices with various speeds standardized by IEC the Fieldbus Foundation chose, a low speed wire, an optical fiber media, and Ethernet.

2.2.1 Physical layer of 31.25 kbps

The physical layer of 31.25 kbps is the most common one since IEC and ISA approved it in 1992. Only a wire medium is explained in this textbook but an optical fiber can also be applied as an alternative. 31.25 kbps may sounds slow in comparison with the latest telecommunication technologies; however, it is needed to satisfy various requirements by replacing traditional 4 to 20 mA analog transmissions. It is designed to apply to field devices used in the industrial plants under various environments. Many users may still prefer two-wired transmitters for its simple installation. High-performance electronics cannot be used in the plant's hazardous area filled with explosive gases. 31.25 kbps transmission speed was chosen for those applications which demand for devices with very low power consumption.

A field device is supplied with electric current from the medium to feed energy to its electronics circuits. It is called a "bus-powered" device, a feidlbus function equivalent to the two-wire device.

2.2.2 Signaling Method

The fieldbus has adopted a technology similar to the one for smart (hybrid) transmitters to transmit electric signals. Figure 2.5 shows an equivalent circuit of signal transmission. Supply voltage is provided by a power supply through an impedance conditioner. DC current through the impedance conditioner feeds devices. Supply voltage is in between 9 and 32 V DC at the device terminals. The impedance conditioner supplies power supply output impedance by 400 ohm or higher in the signal frequency bandwidth.

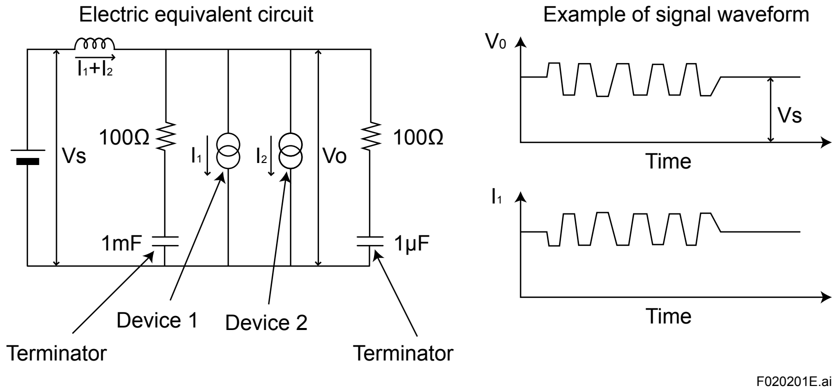


Figure 2.5 Electric Equivalent Circuit of Signal Transmission

A 100-ohm impedance terminator is installed at each cable terminal. It makes an instrumentation cable a balanced transmission line so that a relatively high frequency signal can be transmitted with a minimum distortion.

In Figure 2.5, I_1 and I_2 are the currents flowing through device 1 and device 2. When I_1 of device 1 increases by 10 mA, the voltage between the cables drops by 0.5 V (= 10 mA x 50 ohm (two of the 100-ohm terminators are set in parallel)) as the increased current is supplied from capacitors in the terminators because the inductors in the power supply's impedance conditioner prevent fluctuations of current from the power supply. Then the device 1 decreases I_1 by 20 mA, the voltage between the cables increases by 1 V DC as the current is supplied to the terminator's capacitor. Thus the average current (Vs) is maintained at the same level and generates a modulated signal of 1 V p-p amplitude.

Data is encoded as a voltage change in the middle of the one-bit time (32 μs at 31.25 kbps). The data 1 is encoded as a voltage drop in the middle of the one-bit time while the data 0 as a voltage increase.

Other than these, N+ and N- are encoded as the constant voltage between the bit times such as start/stop signals used for dividing frames to generate specifically identified signals separated from ordinary data. Both N+ and N- are applied for start/stop delimiters encoding the start and the stop of PHL SDU (=DL PDU). The physical layer transmits any combinations of data 1 and 0 in DL PDU.

Figure 2.6 shows a typical waveform of a physical layer signal. The receiving physical layer retrieves the bit time using the preamble and the octets (bytes) of start delimiter signal. The end delimiter indicates the end of the physical layer signal. The length of the preamble can be increased when the signal goes through the repeaters.

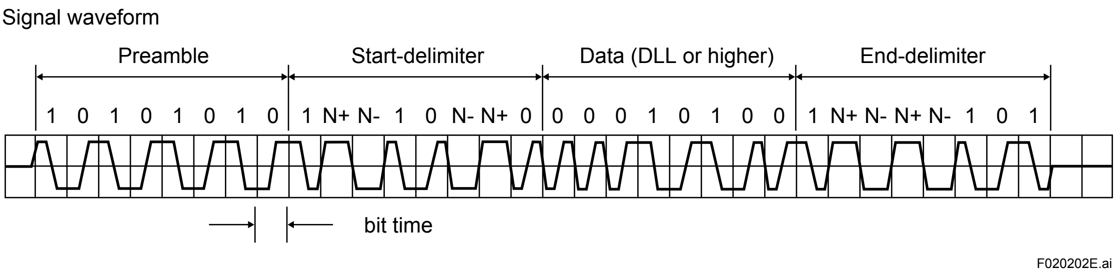


Figure 2.6 Typical Waveform of Physical Layer Signal

2.2.3 Wiring rules

The IEC standards define the minimum amplitude and the worst waveform of a received signal at a device at a place of the fieldbus network. The physical layer receiver circuit must be able to receive this signal.

The transmission line can be configured in any way as long as the received signal quality is guaranteed at all receiving nodes. The ISA SP50 Working Group established a set of wiring rules to simplify the network design. The quality of the received signal always fulfills the minimum requirement under allowable noise level as long as the fieldbus network is designed according to these rules. Although the rules may seem somewhat conservative, they are useful for easily designing a network. In many cases, project set their own wiring rules modified by the project specifics.

Figure 2.7 shows the definitions of trunk and spur cables.

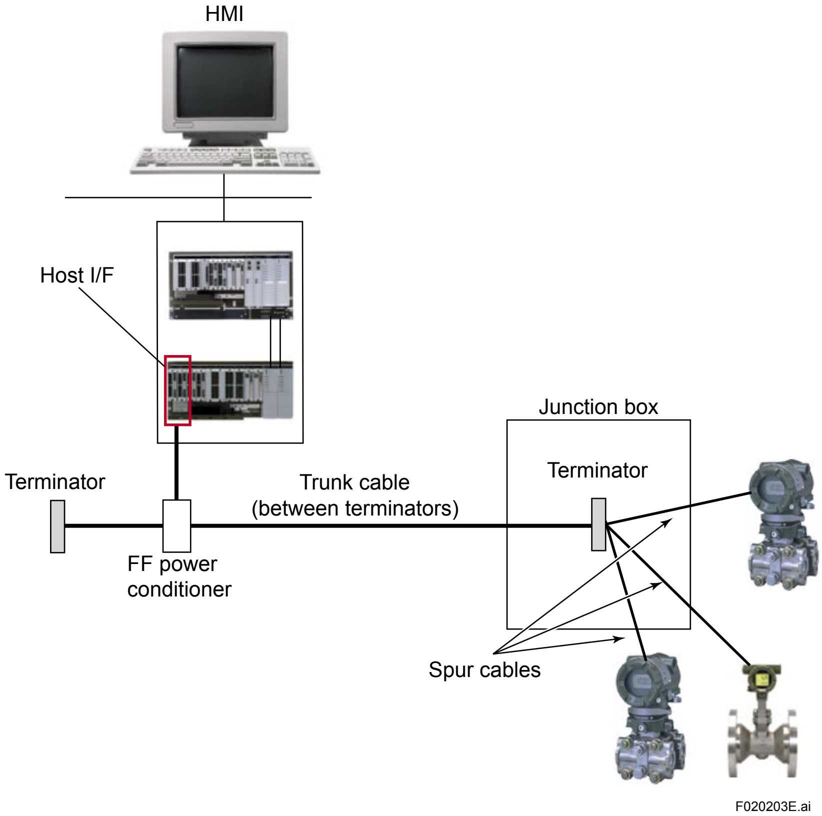


Figure 2.7 Trunk cable and spur cables

Table 2.1 Fieldbus cable types and transmission length

Type of cable	Cable specifications	Max. length of cable (reference value)
Type A: Individually shielded twisted-pair cable	#18AWG (0.82 mm^2)	1900 m
Type B: Multi-pair cable, twisted, with outer shield	#22AWG (0.32 mm^2)	1200 m
Type C: Twisted-pair cable, unshielded	#26AWG (0.13 mm^2)	400 m
Type D: Multi-pair cable, no twist, with outer shield	#16AWG (1.25 mm^2)	200 m

Note: Yokogawa recommends the use of Type A.
Usage of Types B and D is restricted.
Usage of Type C is not recommended.

Table 2.2 Recommended values of maximum length of spur cables (*1)

No. of devices on the fieldbus	Recommended maximum spur length
2 to 12	120 m
13 to 14	90 m
15 to 18	60 m
19 to 24	30 m
25 to 32	0 m

*1: In principle, the maximum spur cable lengths must be set within the lengths shown in table 2.2. However, in an actual application, the maximum spur lengths may not fall within the defined value in some cases. Considering that the lengths defined by the IEC standards as reference values, Yokogawa figured out if the maximum spur length could be extended by using CENTUM production control system, which resulted as follows:

- Type A cable is used as a fieldbus.
- Number of field devices connected per segment: Max. 16 devices
- Maximum spur cable length: 120 m
- Maximum total spur cable length: 1440 m
- Maximum trunk cable length: 1900 m - total spur cable length

There is no restriction in the number of junction boxes used for connecting field devices via spur cables. In the IEC 61158-2 the maximum spur cable length is defined as a reference value and those are all the same for Type A, B, C, and D.

Yokogawa does not recommend fieldbus cables other than Type A or Type B.

By referring to the cable wiring drawing, users are able to confirm if the fieldbus configuration satisfies the above rules. There are other factors to limit the number of devices connected to the fieldbus and those are described in the chapter 2.2.4 of this document.

2.2.4 Intrinsic safety (IS) consideration

Intrinsic safety (IS) installation is important for plants operated in explosive gases atmosphere.

IS is the rules to design and install devices in a hazardous area to prevent an explosive gas being ignited by electric discharge or the surface temperature of a device. An intrinsically safe field device is so designed for preventing ignition even when the component fails.

An IS barrier must be installed to segregate the hazardous area from the safe area as shown in Figure 2.8. The barrier strictly limits the voltage, current, and power fed to a device installed in the hazardous area and a field device must be operational with the limited power supply.

Devices and barriers must strictly comply with the design criteria defined and provided by the safety organizations (IEC, FM, CENELEC, PTB, etc.).

Note that the IS is the only technology applicable to zone 0 where explosive gases always exist. In zone 1 where such gases exist most of the time, explosion-proof technology is also applicable along with IS. The explosion-proof is a technology of housing design and is independent from fieldbus technology.

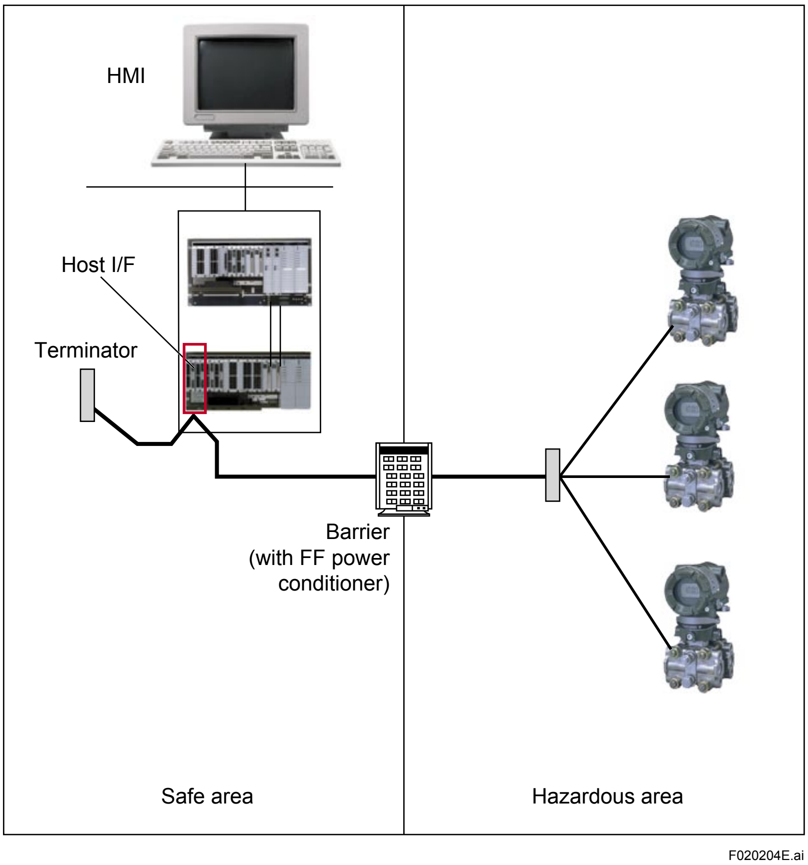


Figure 2.8 Isolation between the safe area and the hazardous area

2.3 Data link layer

Data link layer is a mechanism of transferring data from a node to other nodes that need the data. It also manages priority and orders of such transfer requests. The data link layer concerns data, address, priority, medium access control, and all others related to message transfer. Since the data link layer operates on the low speed physical layer, it has mechanisms to use the medium in an effective way. The FOUNDATION fieldbus data link layer is a subset of IEC61158-3/4 type 1.

2.3.1 Medium access control

The most important functionality of the data link layer is the medium access control (MAC) of the fieldbus. Since all the devices on the single cable receive signals from the same physical layer, only one of them is allowed to transmit its signal at a certain time. MAC makes this happen. The group of devices sharing the same signal in the physical layer. In other words, only one device on a link is allowed to use the medium (physical layer) at a time.

Link active scheduler (LAS) (*1) has a role to control the medium access. “Token” is the right to send a PDU. The LAS possesses the token and passes it to another device to allow it to send messages. The token is then returned to the LAS for further medium access control.

Since some of the application messages have various levels of urgency, the data link layer supports a mechanism to transmit messages in accordance with their urgency. The data link layer provides three levels of “priorities,” which are URGENT, NORMAL, and TIME_AVAILABLE in this order. An URGENT message is transmitted immediately even when other messages of NORMAL or TIME_AVAILABLE are in the waiting queue. The maximum data size allowed for each priority is shown in Table 2.3.

*1: LAS functionality is explained in Chapter 2.3.3.

Table 2.3 Maximum data size by the priority

Priority	Maximum DLSDU data size
URGENT	64 bytes
NORMAL	128 bytes
TIME_AVAILABLE	256 bytes

DLSDU: Data Link Service Data Unit

2.3.2 Addresses

Communication partners in the data link layer are identified by the DL-address consisting of three components - Link, Node, and Selector. Table 2.4 shows an example of bit lengths for Link, Node, and Selector. The link field is composed of 16 bits and identifies a “logical link.” When the communication is closed within a link, this field is often omitted. However, it is needed when a message is transmitted to other links through bridges.

Table 2.4 Bit lengths for Link, Node, and Selector

Data link address component	Bit length example
Link	16 bits
Node	8 bits
Selector	8 bits

Node address is provided in 8-bit length. A fieldbus device has a node address in the ranges from 0x10 to 0xFF, and those are used by categories of LM, BASIC, default, and temporary. Devices are usually in the LM or the BASIC range specified by their device classes. In case a device loses its node address, an address in the default range is used for communication. Temporary devices such as a handheld terminal use node address in the temporary range. The link active scheduler uses 0x04 node address.

Table 2.5 and Figure 2.9 show the address ranges used in a fieldbus link. V(FUN) and V(NUN) are the parameters to determine the area that cannot be used by users. If a device has an address in this area, the fieldbus link does not acknowledge the device.

Table 2.5 Node address range in the fieldbus link by the device class

Node address range in fieldbus link	Device class
0x10 to V(FUN)	Address for Link Master (LM) class devices
V(FUN) + V(NUN) to 0xF7	Address for BASIC class devices
0xF8 to 0xFB	Default address for devices with cleared address
0xFC to 0xFF	Address for temporary devices like a handheld communicator

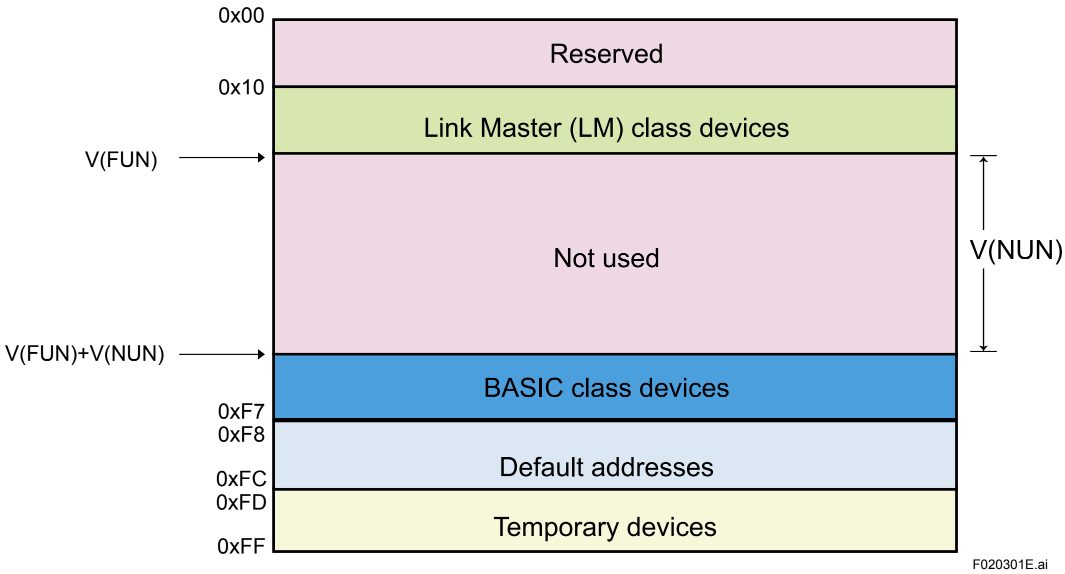


Figure 2.9 Node address ranges in the fieldbus link

The selector field is an 8-bit device internal address for identifying a VCR. When the VCR is connected to another VCR, it is identified by the data link connection end point (DLCEP) address shown in this field.

When the VCR is open to send/receive messages without being connected with others, it is identified by the data link service access point (DLSAP) address shown in this field. DLCEP and DLSAP have different ranges.

Several of the DL addresses are reserved for specific purposes. For example, devices can share the same "global" DLSAP for receiving alarms.

2.3.3 Link active scheduler

Link active scheduler (LAS) has a role of controlling the medium access. FOUNDATION fieldbus devices are categorized in classes as BASIC, Link Master (LM), and Bridge. A LM class device has a capability to work as the LAS, while BASIC class devices do not. A Bridge class device has, in addition to LM capability, the functionality to connect links.

One and only one device in a link works as the LAS at a certain time. Therefore at least one LM (or Bridge) class device is needed in a link. LM devices try to acquire LAS role when no LAS exists on start up or when the current LAS fails. The LM device with the least node address wins this contention. Other LM devices observe the LAS activity and take over its role when LAS goes away. Figure 2.10 shows the procedure through which a Link Master class device becomes the LAS.

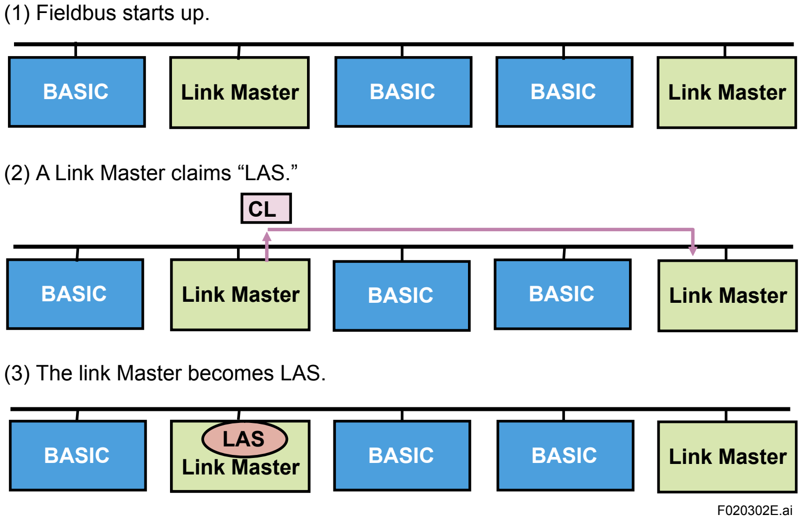


Figure 2.10 The process of LA class device becoming LAS

Note that the LAS has an additional function other than the basic communication. Therefore it has a DL-address (0x04) different from the node address.

2.3.4 Scheduled communication

LAS is responsible for synchronizing scheduled communication. The FOUNDATION fieldbus function block (FF function block) is an application operating in a synchronized manner with LAS.

The LAS manages the communication of the synchronized data transfer.

One of the output parameters of the FF function block is a data "publisher" of data and other FF function block input parameter to receive data is called a "subscriber." The LAS controls periodic data transfer from the publisher to the subscriber by using the network schedule.

When the scheduled communication time comes, the LAS sends compel data (CD) PDU to the publisher DLCEP. Then, the publisher immediately transmits the data transfer (DT) PDU stored in the DLCEP data buffer. When the LAS sends a CD to a publisher, the subscribers are notified that the publisher is to send the data to subscriber. Received data is stored in the subscriber's buffer. The CD PDU, in this case, is the token from the LAS to the publisher and the DT PDU is the token returned from the publisher to the LAS. See Figure 2.11 for details.

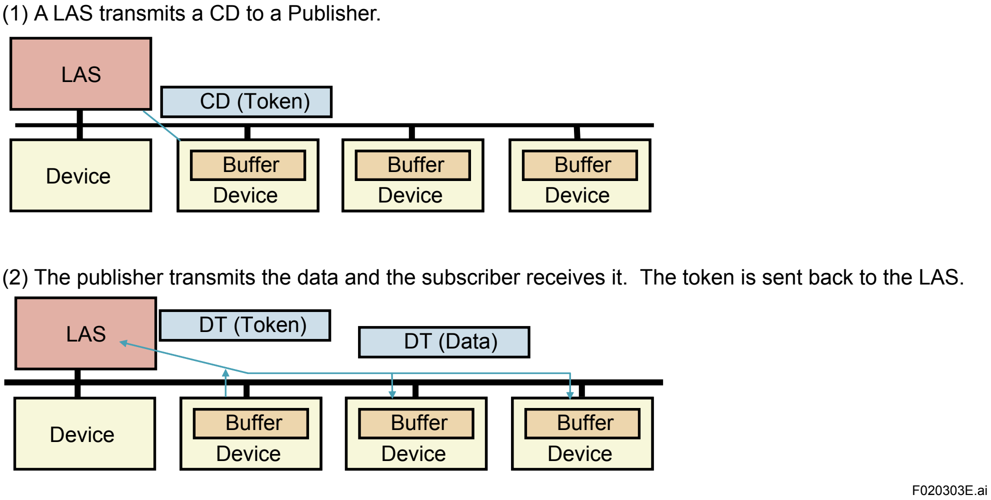


Figure 2.11 LAS control in the scheduled communication

The data link layer appends "freshness" information as a PCI to the data so that the subscribers knows if the data has been updated since its last publishment.

2.3.5 Unscheduled Communication

Other than the scheduled communication, there are asynchronous communications. The LAS is responsible for giving all the nodes on a link chances to send messages.

The LAS gives a token to a node by sending a Pass Token (PT) PDU. A PT PDU contains priority and time interval information. The device holding the token sends the data to another device. When the node does not have messages of the given or higher priority to be sent, or the given-time interval is expired, it returns the token as a Return Token (RT) PDU.

The LAS controls the message transfer by updating the priority. When the token is given to all the devices with a short time interval, the LAS gives more time to the nodes by lowering the priority. When the token cannot reach all devices within the network parameter called a “targeted token rotation time,” the LAS increases the priority so that the token is given to all the devices within the desired time interval.

The device must return the token within the time interval given by the PT PDU. The unscheduled communication must be finished before the next scheduled communication begins.

Note that the token is given to the node and not to DLCEP or DLSAP. Therefore the device has to be responsible for enabling all the DLCEPs and DLSAPs in the device to send messages. See Figure 2.12.

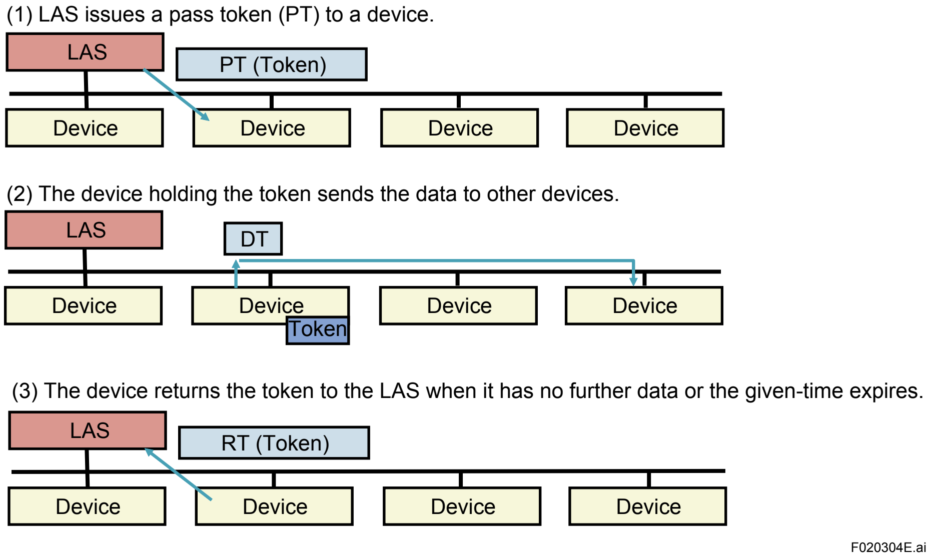


Figure 2.12 LAS control in the unscheduled communication

2.3.6 Link Maintenance

The LAS's third role is to maintain the link. The LAS gives the token to all the devices detected by the LAS. When a new device is added to the network, it must be recognized by the LAS and enlisted into the "live list," a token rotation list."

The LAS sends a probe node (PN) PDU to a node address which is not listed in the "live list" and in the polling address range. The new device waits until it receives PN and returns a probe response (PR) PDU to the LAS afterwards. Then the LAS adds this device to the live list after activating the full DLL functionality of the device. (This activation procedure is beyond the scope of this tutorial.)

This probing process is repeated in a given interval.

When a device is removed from the link, it does not respond to the PT any more. The LAS detects it and deletes the device from the live list.

Whenever changes in the live list are detected, the LAS broadcasts the changes so that all the LM devices share the latest list and are ready to take over the LAS function.

The LAS also broadcasts its data link time (link scheduling (LS) time) using the time distribution (TD) PDU to all the devices connected to the network in a predefined interval. In this way, all the devices on the network share the same time stamp to start the FF function block, thus it is also called "network time."

2.3.7 Data Link PDUs

Table 2.6 summarizes the data link protocol data units (DL PDUs) in for the FOUNDATION fieldbus.

Table 2.6 data link protocol data units (DL PDUs)

DL PDU	Name	Functionality
EC	Establish Connection	Connect DLCEP.
DC	Disconnect Connection	Disconnect.
CD	Compel Data	Poll a Publisher.
DT	Data Transfer	Send a data unit.
PT	Pass Token	Give the token.
RT	Return Token	Return the token.
RI	Request Interval	Request more PT.
PN	Probe Node	Search new node.
PR	Probe Response	Join the link.
TD	Time Distribution	Synchronize Time.
CT	Compel Time	Request TD.
RQ	Round-trip Time Query	Measure delay in TD.
RR	Round-trip Time Response	
CL	Claim LAS	Becomes LAS.
TL	Transfer LAS	Request LAS role.
IDLE	Idle	No activity

2.4 Application layer

The application layer consists of two sublayers. Fieldbus Access Sublayer (FAS) manages data transfer while Fieldbus Message Specification (FMS) encodes and decodes user data.

2.4.1 Fieldbus access sub-layer

Fieldbus access sub-layer (FAS) is a part of an established communication. Since the fieldbus does not have #3 to #6 layers between the DLL and the APL, the FAS directly maps the APL requests to the DLL services. This is the most important part of VCR management.

The FAS provides applications three communication models as shown in the Table 2.7 below. A network manager is expected to correctly configure VCRs in accordance with the models having communications with each other. Once those are configured, the FAS provides the communication methods based on the models.

Table 2.7 Communication models in the fieldbus access layer (FAS)

Model	DLL	Schedule by	Direction
Client/Server	Queued	User	Bi-directional
Publisher/Subscriber	Buffered	Network	Uni-directional
Source/Sink	Queued	User	Uni-directional

Client/server model

The client/server model is generally adopted in many communication technologies. A “client” application sends a request to perform a specific action through FMS to a “server” application.

When the server accomplishes the requested action, its result is sent to the client. It is a one-to-one bi-directional communication using DLCEP.

A typical example is a human-machine interface (a client) to read data of a FF function block (a server).

The client sends a read request to the server and then the server sends the data back to the client. This communication takes place at all times.

The client may send many requests at a time. The client/server VCR forms a queue to save those requests and sends them one by one when the targeted node has a token. A flow control mechanism is applicable for to error recovery and server’s performance managements. See Figure 2.13.

The features of Client/server communication model are as follows.

- One-to-one communication with bi-directional connection
- General communication
    - Manual operation
    - Configuration downloading
    - Maintenance
- Response time depends on
    - Device throughput
    - Bus traffic

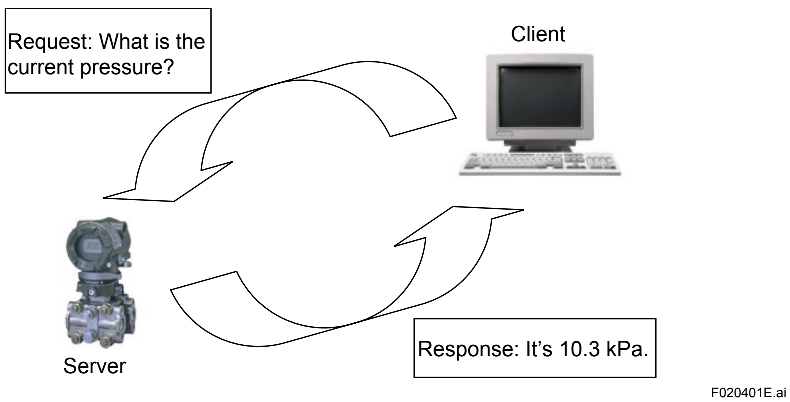


Figure 2.13 Client/server communication model

Publisher/subscriber model

Publisher/subscriber model is designed to transmits device data among FF function blocks. When the FF function block initiates a publishing, its output data is stored in the publisher VCR buffer. Then the LAS sends a CD to the VCR to let it transfer the data in the DT PDU.

The subscriber VCR receives this PDU and provides the data to the subscribing FF function block.

A typical example is to connect an output from an analog input (AI) block with a process value input of a PID control block.

The features of Publisher/subscriber communication model are as follows:

- One-to-many communication with one-way connection
- Data transfer for instrumentation
  - Connecting FF function blocks
- Periodic schedule communication with highest priority

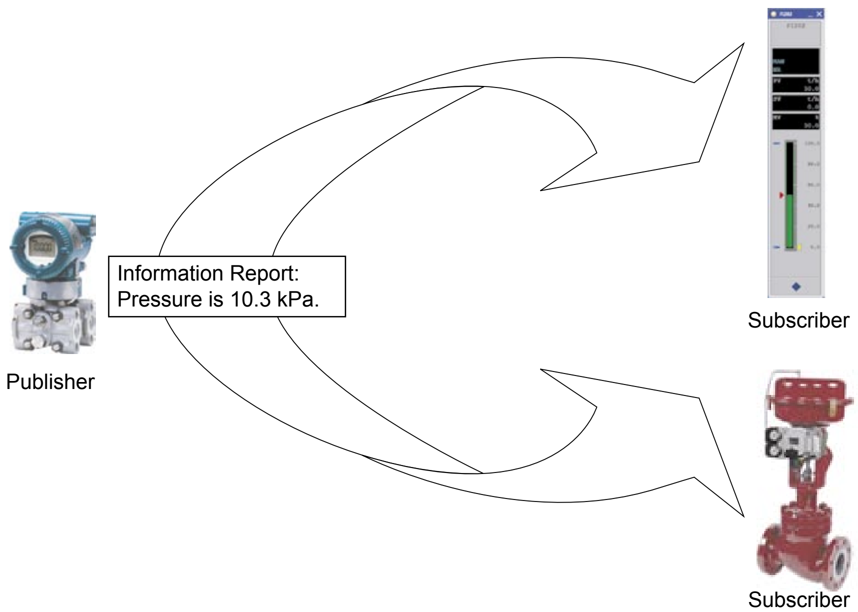


Figure 2.14 Publisher/subscriber communication model

The publisher/subscriber model is a one-way communication for one-to-many using DLCEP. The subscribers are notified whether data has been updated since the last publishment.
With this feature, the publisher, the FF function block, is able to update the buffer at the end of its block execution whether the previous data is sent to the subscriber or not.

Source/sink model

The source/sink model is designed for broadcasting messages. It is a one-way communication for one-to-many without schedules. It is also known as a "report distribution model."

A source VCR transfers a message in the message queue to an assigned global DLSAP address when the device has a token. Several of the sink VCRs possess the same global address and receive the same message from a source.

FOUNDATION fieldbus devices use this model for two specific purposes. One is to notify alarms and events detected in the source, and the other is to transmit the source function block trend data. Alarms and events are acknowledged through a client/server VCR

For an alarm logging function, a sink VCR receives alarms from a source VCR. The sink VCR can receive messages from many sources if those sources are configured to send messages to the same global address. The sink VCR can be identified by DLSAP address of the source.

The features of the source/sink communication model are as follows:

- One-way communication for one-to-many without connection
- Event notification
  - Device alarms or process alarms
  - Trending
- On-demand, non-periodic communication

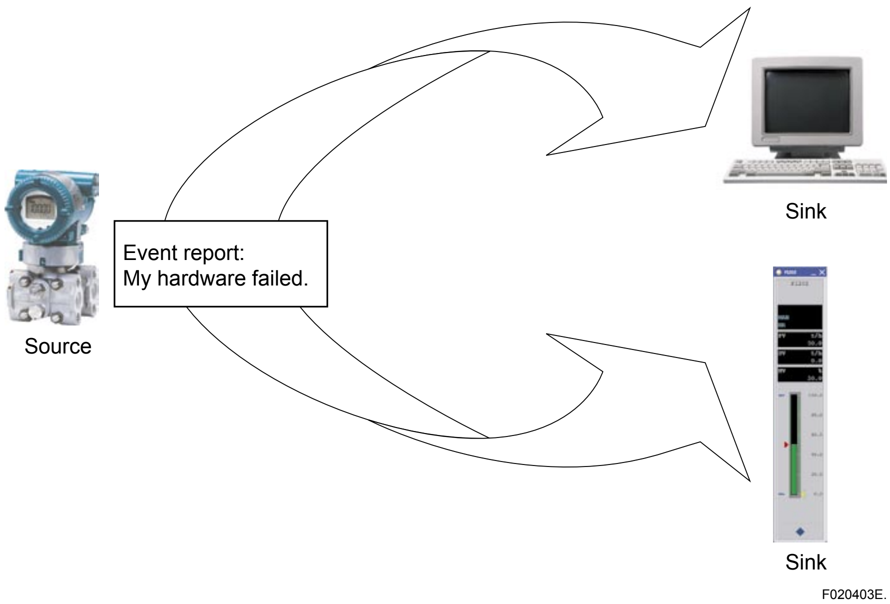


Figure 2.15 Source/sink communication model

2.4.2 Fieldbus message specification

Fieldbus message specification (FMS) is a service interface for user applications to use fieldbus services. FMS encodes a service request upon receiving such demand and transfers it to other applications. The receiver FMS decodes the request and notifies to the application.

■ Virtual field device (VFD)

A fieldbus device may have user applications, which are independent from and not interfering each other. The fieldbus device has virtual field devices for such individual applications. An identifier, given to a VCR, identifies the VFD.

A fieldbus device posesses at least two VFDs. One is a management VFD where network and system management applications reside, which is used for configuring the network parameters including VCRs as well as to manage devices in the fieldbus system.

The other is an FF function block VFD where the FF function blocks exist. A field device can possess two or more FF function block VFDs.

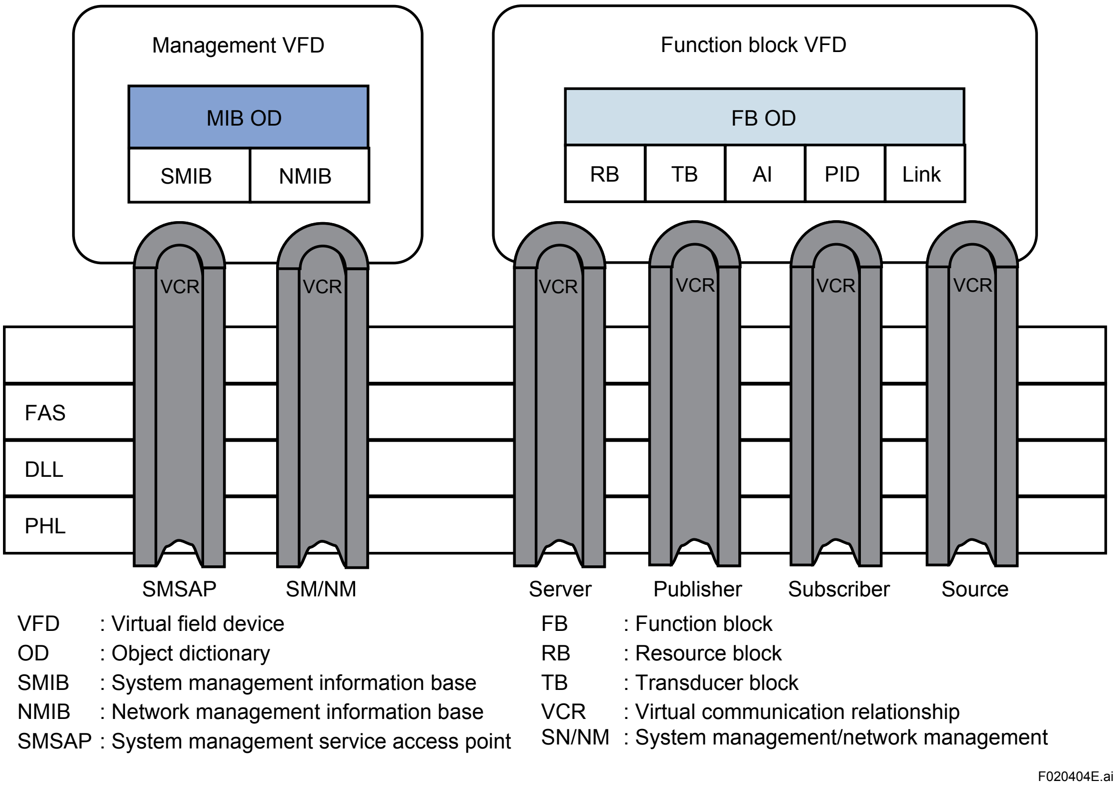


Figure 2.16 Management VFD and FF function block VFD

FMS Objects

Applications in a VFD are shown to other applications on the network using an "object model," which consists of attributes, its behavior and services.

Object Examples

The FF function blocks have parameter objects accessible from other applications. Alarms, FF function block linkage are also the objects. Their behaviors are specified in the FOUNDATION fieldbus function block application specifications.

The network behavior is managed via the network management information base (NMIB) objects. The system behavior is managed via system management information base (SMIB) objects. Schedules and VCRs are also classified as objects.

Object Dictionary

An object is identified by a unique number in the VFD called "index." Additional information to describe an object is necessary for configuring an open systems, which is called the "object dictionary (OD)." The "object description" is a collection of object information to describe objects.

A client application can read such explanations using "Get OD" service and read the value when the object is a variable.

The most fundamental object is a "variable" containing a value. It may be a simple variable, a record (structure) or an array. FF function block parameters, VCR, NMIB, and SMIB are examples of the record variables.

Event, domain, and program are also categorized as objects. These are described in the following section. Figure 2.17 shows relationship of the objects and object descriptions.

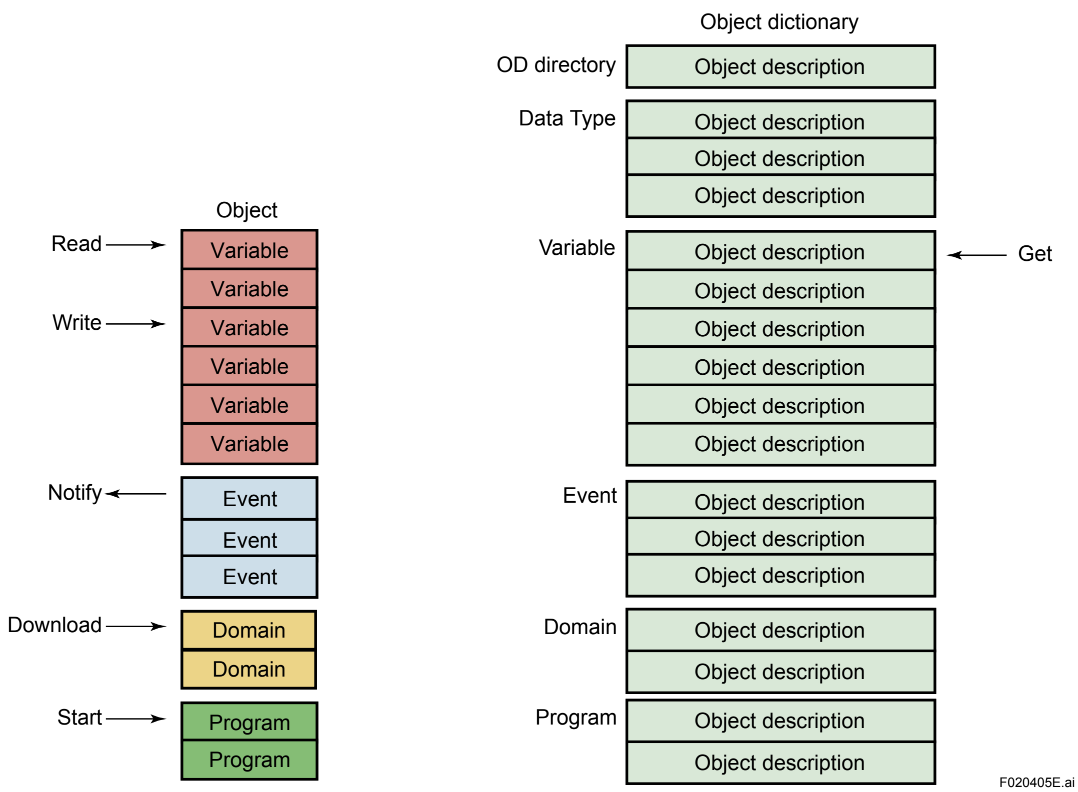


Figure 2.17 Objects and object descriptions

Object descriptions in the OD may support an extension attribute. This extension is used by the function block application to support object naming, OD reference, and so on. The DD reference is the pointer to the Device Description corresponding to the object and enable client applications to handle the object according to the Device Description.

FMS services

FMS provides users services to access FMS objects. Table 2.8 to 2.11 show summaries of the services by the service classes.

Variable access

A variable is a place where a data is stored. Its value can be read or written by other applications. A variable data can be transmitted to other applications using the information report service without any request. The FOUNDATION fieldbus uses this service in data publishing and trend reporting. For effective data transfer, a list of variables can be defined as "variable list."

When a variable, composed of multiple variables, is a record or an array all or only one component assigned by the "sub index" can be sent.

Table 2.8 Variable Access Services

Service	Functionality
Read	Read a variable
Write	Write a variable
InformationReport	Send data
DefineVariableList	Define a variable list
DeleteVariableList	Delete a variable list

Event management

An event is used for notifying that an application has detected something important. Failure, data update, and alarms are examples of events. The event is reported repeatedly in the source/sink model until it is acknowledged by the client/server model. Its notification can be enabled or disabled via other event-related services.

Table 2.9 Event services

Service	Functionality
EventNotification	Report an event
AcknowledgeEventNotification	Acknowledge an event
AlterEventConditionMonitoring	Disable/enable event

Domain Management

A continuous memory area is referred to as a domain which is a program area or a data area. A client can download a data to a domain or upload a domain content via FMS services.

Since the domain size can be enlarged than the maximum size of the FMS encoding, the FMS allows uploading or downloading a domain by dividing it up in smaller sections. Initiation and termination services are provided for managing a partial download and upload.

Table 2.10 Domain services

Service	Functionality
RequestDomainDownload	Request download
(Generic) InitiateDownloadSequence	Open download
(Generic) DownloadSegment	Send data to device
(Generic) TerminateDownloadSequence	Stop download
RequestDomainUpload	Request upload
InitiateUploadSequence	Open upload
UploadSegment	Read data from device
TerminateUploadSequence	Stop upload

Program invocation services

A data processing function to be managed from other applications is referred to as a program. It is designed for PLC ladder programs and can be used for FF function block applications.

When downloaded a program, its invocation is tightly coupled with the domain management.

Table 2.11 Program Services

Service	Functionality
CreateProgramInvocation	Create a Program object
DeleteProgramInvocation	Delete a Program object
Start	Start a program
Stop	Stop a program
Resume	Resume program execution
Reset	Reset the program
Kill	Remove the program

Other services

The FMS provides the object dictionary and the context management.

The object description can be read by the “GetOD” service. Upon downloading an object, its object descriptions need to be downloaded as well.

The “context” management services are used to establish a connection, to release a connection, and to reject improper services.

Table 2.12 List of FMS services

	Service	Functionality
OD management services	GetOD	Read an object dictionary
	InitiatePutOD	Start an OD load
	PutOD	Load an OD into a device
	TerminatePutOD	Stop downloading of OD
Context management services	Initiate	Establish communications
	Abort	Release communications
	Reject	Reject improper service
	Status	Read a device status
	UnsolicitedStatus	Send unsolicited status
	Identify	Read vendor, type and version

2.5 System management protocol

The “System Management Kernel Protocol (SMKP)” uses a data link layer service without an application layer, which is operational even under abnormal situation such as system startup, wrong configuration, device failure, and device replacement.

2.5.1 Tag and address assignment

A field device is identified with its PD tag and its node address as explained in Chapter 3.4.1 “Device Management.” The SMKP provides a service to assign them to a device.

2.5.2 Tag location

The device tag and the FF function block tag are convenient tools for operators; however, those can be as short as possible for the effectiveness of the fieldbus communication. SMKP provides services to replace a device tag and a block tag with a node address and an index for simplifying communications.

2.5.3 Application time synchronization

The fieldbus applications need time synchronization as those interacts each other. For example, an event message needs a time stamp to indicate when it is detected, because it is received delayed due to the token rotation and bus traffic. The SMKP provides a mechanism for all the management VFDs to share the synchronized time.

3. Fieldbus Applications

This chapter explains the FOUNDATION fieldbus function block application and other applications running on the FOUNDATION fieldbus. Communication technologies supporting those applications are explained in Chapter 2 of this document.

In this chapter, how control and measurement applications are implemented on the FOUNDATION fieldbus.

Furthermore, how carefully those applications are designed to provide secure control and the successful maintenance to a plant is described.

3.1 Virtual field devices

A fieldbus device can have user applications which are independent from each other with no interaction. Fieldbus devices are composed of virtual field devices (VFDs) to accommodate with such applications

From the application viewpoint, VFDs can be regarded as different field devices. Communication services guarantee VFDs independence. See Figure 2.16 “Management VFD and FF function block VFD” for relationship between the VFD and communication.

3.1.1 VFDs in a fieldbus device

A FOUNDATION fieldbus device has two or more VFDs.

One is the “Management VFD” where network and system management applications reside. It manages the devices on the fieldbus as well as for configuring network parameters including VCRs.

The other is a “FF function block VFD” where FF function blocks exist. A field device can have two or more FF function block VFDs.

3.2 FOUNDATION fieldbus function block

This section focuses on the FOUNDATION fieldbus (FF) function block, the most important concept in the FOUNDATION fieldbus, especially its models and parameters. Users are to configure applications, maintain, and customize their applications through the FF function blocks.

3.2.1 What is an FF function block?

An FF function block is a common functional model in control and measurement. It is a generalized concept of the functionality used by field instruments and process control systems such as analog input and output as well as PID control. "Function Block Application Process - Part 1," the FOUNDATION fieldbus specifications, describes the fundamental concepts of the FF function block while "Function Block Application Process - Part 2" and the sequels explains details of the various FF function blocks.

The FF function block parameters are accessed and monitored via communication services and the block behavior depends on the parameter values. The FF function block can be resided in any device on the network and a set of FF function block connected to each application can be resided in one device or distributed among devices. The Fieldbus Foundation's system architecture document describes it as follows:

> "One of these models, the function block model, has been specified within the architecture to support low level functions found in manufacturing and process control. Function Blocks model elementary field device functions, such as analog input (AI) functions and proportional integral derivative (PID) functions.
>
> The function block model has been supplemented by the transducer block model to decouple function blocks from sensor and actuator specifics. Additional models, such as the 'exchange block' model, are defined for remote input/output and programmable devices.
>
> The function block model provides a common structure for defining function block inputs, outputs, algorithms and control parameters and combining them into an Application Process that can be implemented within a single device. This structure simplifies the identification and standardization of characteristics that are common to function blocks."

The blocks in the FF function block VFD are classified in three blocks: resource block, FF function block, and transducer block.

■ Resource block

The resource block indicates things existed in the VFD. It contains the manufacturer's name, device name, DD revision and so on. The resource block controls overall device hardware and FF function blocks within the VFD.

■ FF function blocks

The FF function block is a generalized model for control and measurement. For example, the AI block adjusts raw data from transducer(s) and outputs the measured value in a common format.

The FF function block is categorized in three classes:

(1) Standard block as specified by the Fieldbus Foundation,

(2) Enhanced block with additional parameters and algorithm, and

(3) Open block or a custom block designed by each manufacture.

The FF function block consists of input, output and contained parameters. The data generated in the block is regarded as an output parameter, which can be linked to the input parameter of other FF function blocks.

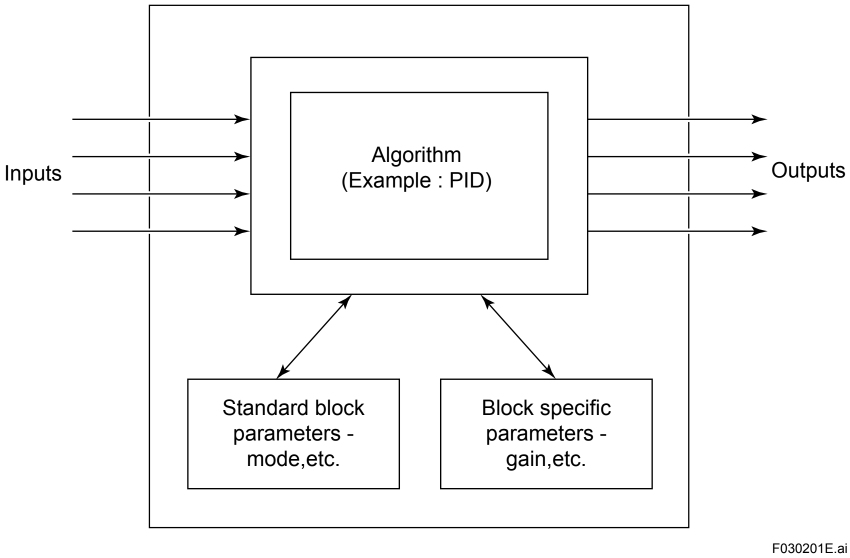


Figure 3.1 An example of the FF function block (PID block)

Table 3.1 lists the FF function blocks defined by the Fieldbus Foundation. The major functions are implemented here. “Function Block Application Process - Part 2” describes about the most fundamental function blocks for control and measurement. Part 3 of the document describes about the advanced control blocks. Part 4 is for multiple I/O blocks for interfaces for other devices such as manufacturing automation ones other than the FOUNDATION fieldbus devices. Part 5 shows how to develop users’ applications, like a PLC program, in the way it is required to be interoperable with the standard FF function blocks.

Table 3.1 Function block list

FIELDBUS FOUNDATION TECHNICAL SPECIFICATIONS NUMBER	SYMBOL	FUNCTION BLOCK NAME
Function block AP Part 2 (*1) - Basic control block	AI	Analog input
	AO	Analog output
	BG	Bias/gain
	CS	Control selector
	DI	Discrete input
	DO	Discrete output
	ML	Manual loader
	PD	Proportional/derivative control
	PID	Proportional/integral/derivative control
	RA	Ratio
Function block AP Part 3 (*2) - Advanced control block	DC	Device control
	OS	Output splitter
	SC	Signal characterizer
	LL	Lead lag
	DT	Deadtime
	IT	Integrator (Totalizer)
	SPG	Setpoint ramp generator
	ISEL	Input selector
	AR	Arithmetic
	TMR	Timer
	AAL	Analog alarm
	TOT	Totalizer
Function block AP Part 4 (*3) - Multiple I/O block	MAI	Multiple analog input
	MAO	Multiple analog output
	MDI	Multiple discrete input
	MDO	Multiple discrete output
Function block AP Part 5 (*4) - Flexible function block	FOD	Fixed OD
	FPR	Fixed programmable resource
	VOD	Variable OD
	VPR	Variable programmable resource

*1: For details, refer to Technical Specifications No. FF-891 issued by the Fieldbus Foundation.
*2: For details, refer to Technical Specifications No. FF-892 issued by the Fieldbus Foundation.
*3: For details, refer to Technical Specifications No. FF-893 issued by the Fieldbus Foundation.
*4: For details, refer to Technical Specifications No. FF-894 issued by the Fieldbus Foundation.

Various control and measurement applications can be configured by linking these function blocks. Figure 3.2 shows typical examples using basic control blocks.

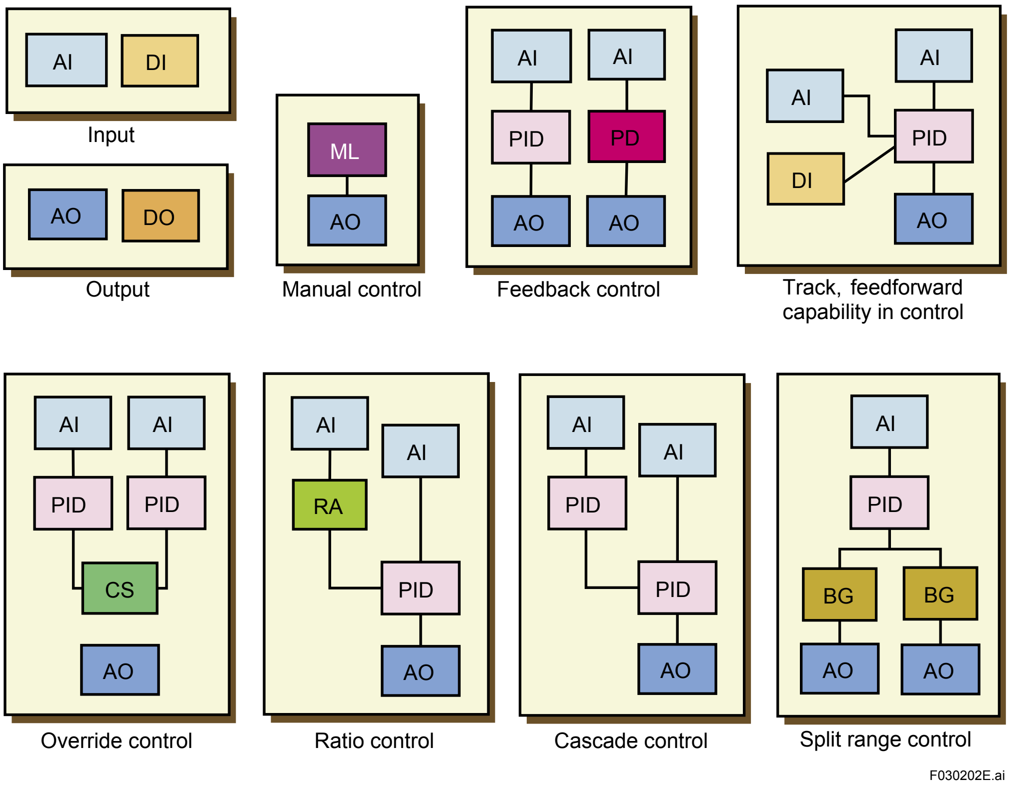


Figure 3.2 Examples of function block linkage using basic control function blocks Introduced in the "Function block AP Part 2."

Transducer block

The transducer block represents models of sensors and actuators. It is designed similar to the expressions of FF function blocks. Sensors in the pressure transmitters can be mapped on the transducer block. The transducer block links to the FF function block via the CHANNEL parameter of the FF function block.

Contrary to the FF function block being a generic concept, the transducer block is dependent on its hardware and measurement principles. For example, a pressure transmitter and a magnetic flowmeter have different measurement principles but both provide analog measured values. The common part is modeled as an analog input (AI) block. The different part is modeled as a transducer block that provides the information on the measurement principle.

The Fieldbus Foundation is developing standardized transducer model for each sensor and actuator type for further interoperability and ease of use.

- Pressure transducer block
- Temperature transducer block
- Positioner transducer block

3.2.2 Function block linking and scheduling

A control or measurement application consists of FF function blocks connected to each other. Figure 3.3 shows an example of PID control loop composed of AI, PID and AO blocks. These blocks are connected via “link objects” in FF function block VFD. A link object is to connect two FF function blocks within a device, or an FF function block with a VCR for a publisher or subscriber.

An FF function block must obtain input parameters prior to the execution of its algorithm. Its output parameters must be made available after the execution of the algorithm.

Therefore algorithm execution and the publisher-subscriber communication must be synchronized even when the FF function blocks are distributed among devices.

The system management associates with the data link layer to achieve this synchronization by using the link scheduling (LS) time distributed and synchronized by the link active scheduler (LAS).

The system management of a field device triggers FF function block according to the FF function block schedule. The LAS, in accordance with the LAS schedule, transmits the compel data (CD) PDU to a publishing device to urge the output data to be transmitted. These two schedules (FF function block schedule and LAS schedule) are defined as offsets (orders) in the “macrocycle,” a control period. Then the FF function block and the communication must be scheduled as per the offsets. See Figure 3.3 as an example.

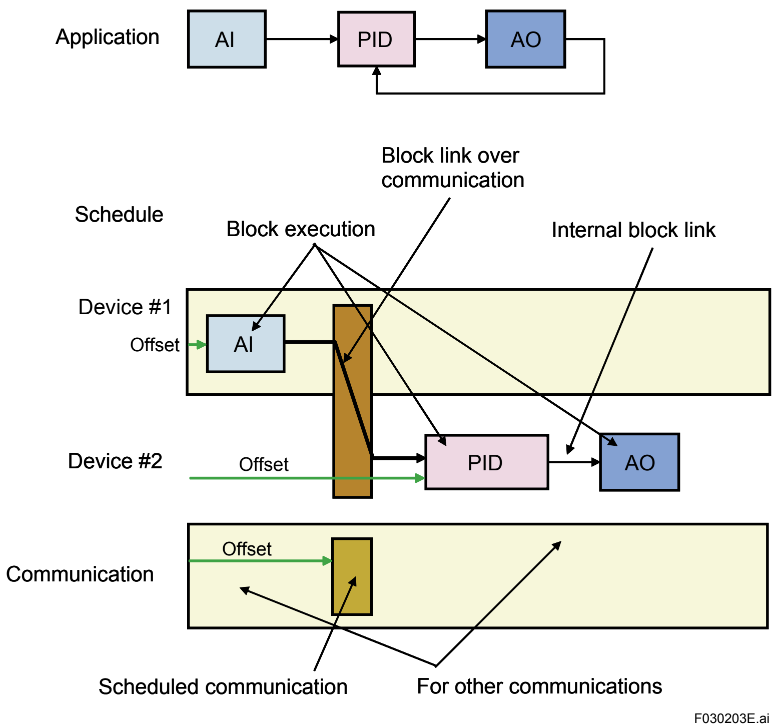


Figure 3.3 Example of scheduling FF function blocks and communications

3.2.3 Parameters

The FF function block has various parameters which are accessible by selecting one of the FMS index. The FF function block parameters have continuous indices.

■ Parameter Classes

The block parameters are categorized into three classes: input, output, and contained parameters. The FF function block has all of them while the resource block and transducer block have only contained parameters.

● Output Parameters

An output parameter is an output of the FF function block and can be connected to input parameter(s) of other function block(s). One FF function block parameter can be connected with two or more FF function block INPUT parameters. This is supported by the periodic publisher/subscriber communication. An output parameter is a record consisting of a value (analog or discrete) and its status (showing whether the value is applicable).

● Input Parameters

An input parameter is an input of the FF function block and it accepts an output parameter of other FF function block. Its data type must be the same type as the output parameter.

● Contained Parameters

A contained parameter is neither input nor output. It is accessible only through on-demand read or write request. Any of the data types defined by the Fieldbus Foundation can be used.

Parameter Attributes

Block parameters have several attributes that creates parameter's complex behaviors.

Access right

The FF function block parameters are readable (expressed as "r") and/or writable (expressed as "w"). However, even when a parameter is writable, there are certain restrictions. For example, an OUT parameter in an AI block is writable only when the block mode is O/S (out of service) or MAN (manual). In other block modes, a write request to this parameter is rejected. The vendor-specific range check for a write request is allowed. For example, most of the PID blocks reject such requests to set proportional gain to zero.

Dynamic or static

Dynamic parameter is calculated by the block algorithm and therefore does not need to be restored after a power failure. Static parameter has a specific configured value that must be restored by a device after power failure. Non-volatile parameter is written on a frequent basis and the last saved value must be restored by a device after power failure. Since the values of dynamic parameter attributes are constantly changing, they are not normally tracked by a configuration device.

Parameter status

The Input parameters, the output parameters, and several of the contained parameters are available with status.

These parameter qualities are shown in the status. The status is GOOD when the value is usable and BAD when it is not usable. The status can be UNCERTAIN when the block applicability of the value is not 100% guaranteed.

The FF function blocks interpret parameter status of UNCERTAIN as in the same manners as GOOD or BAD. The parameter status has additional fields to show more details.

3.2.4 Important parameters

Several parameters are common to the FF function blocks and those plays important roles in engineering.

■ Block mode

All blocks have their own MODE_BLK parameters which have four elements: target, actual, permitted, and normal modes. The target mode is the records of block modes that an operator wishes to bring into, and its elements are writable.

The actual mode shows the block's actual mode and is read only. When the required conditions match, the actual mode becomes as the same as the target mode, unless otherwise, the actual mode stays different from the target mode for some reason.

The permitted mode shows which mode is allowed in the FF function block's target. The normal mode records and guides the mode that the operators expect under the normal conditions. Both permitted mode and normal mode are writable but it is not a good idea to change them without a reason.

O/S, MAN, AUTO, CAS, RCAS and ROUT modes are available. In O/S (out of service) mode, the block always sets parameter status to BAD. In MAN (manual) mode, its output is not influenced by the FF function block execution. In AUTO (automatic) mode, the block functions independently from the FF function blocks upstream. In CAS (cascade) mode, the FF function blocks receives a setting value from the FF function blocks upstream. The FF function block's actual mode for output or control may become IMAN (initialize manual) or LO (local override) according to the status of a downstream block or local operation.

The decimal point element specifies the number of digits to the right of the decimal point that should be used by an interface device in displaying the specified parameter.

■ Scaling parameters

Several of the function blocks need data scaling of 0 to 100%. A scaling parameter is a record consisting of four elements: EU@100%, EU@0%, units index and decimal point. EU@100% and EU@0% are values of 100% and 0%, respectively, in the engineering units. The units index indicates the engineering units of scales or parameters such as kg/s, MPa, and m, psi, inches or something else.

The decimal point element specifies the number of digits to the right of the decimal point that should be used by an interface device in displaying the specified parameter.

3.2.5 View objects

Since the FF function block contains many parameters, it is not practical for operators to read them one by one for display purposes. The Fieldbus Foundation has adopted the concept of the FMS variable list for the convenience and simultaneity of the set of parameter values. View objects are predefines groupings of block parameter sets, and four view objects are assigned to each block.

VIEW_1: Operation dynamic – information required by a plant operator to run the process.

VIEW_2: Operation static – Information which may need to be read once and then displayed along with the dynamic data.

VIEW_3: All dynamic – Information which is changing and may need to be referenced in a detailed display.

VIEW_4: Other static – Parameters mainly for configuration and maintenance information. Not all of the static parameters are in the VIEW_4.

3.3 Important function blocks

The Fieldbus Foundation specified ten (10) standard function blocks in its FOUNDATION™ Specification Function Block Application Process Part 2. More function blocks are specified in the FOUNDATION™ Specification Function Block Application Process Part 3, Part 4, and Par 5. Out of many function blocks, only five (5) of them (AI, DI, PID, AO and DO) are important, especially the three (3) of them which are AI, PID, and AO.

The information of three important function blocks (AI, AO and PID) as well as resource and transducer blocks are described in this section. Here, Yokogawa offers "recommended" values of the block parameters for users to use based on the long experience in control and measurement.

3.3.1 AI block

The analog input (AI) block is designed to let users use "standard" models of the generalized signal-conditioning function. The AI block receives and processes data measured by the transducer block as follows:

- Scaling
- Square-root calculation (for an orifice plate)
- Low-pass filter
- Alarm generation

Figure 3.4shows the internal stucture of the AI block.

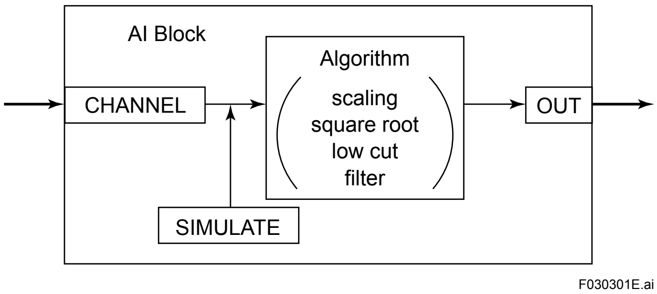


Figure 3.4 AI Block

Scaling and square root calculation

Parameter L_TYPE (linearization type) controls scaling and square root. The process value (PV), a contained parameter of the AI, is determined by the L_TYPE. When its value is “direct,” the channel value becomes OUT value. When it is “indirect,” the channel value is scaled with XD_SCALE and OUT_SCALE.

XD_SCALE provides 0% and 100% value of the channel value and its engineering unit, while OUT_SCALE provides those of the output value. 0 – 100% of the channel value is mapped to 0 - 100% of the output value.

When the L_TYPE value is in “Square Root,” the output value is square root of the scaled value. The calculated value is not unstable due to the nature of orifice plate’s characteristics. In that case, a cutoff function is applied to force the PV to zero when it is lower than the LOW_CUT value.

Low-pass filter

PV can be better stabilized by applying single-exponential low-pass filter. Its time constant is given by PV_FTIME in the units of a second. When its value is zero, no filter is applied.

Alarm generation

When the PV is lower than LO_LIM or LO_LO_LIM, a LO or LO_LO alarm is generated respectively. When the PV is higher than HI_LIM or HI_HI_LIM, a HI or HI_HI alarm is generated, respectively. Alarms are usually generated in the following orders.

LO_LO_LIM ≤ LO_LIM ≤ HI_LIM ≤ HI_HI_LIM

Permitted mode

The permitted mode consists of O/S, MAN, and AUTO. In MAN mode, OUT.VALUE can be modified by users. In AUTO mode, PV.VALUE and PV.STATUS are copied to OUT.VALUE and OUT.STATUS respectively.

Channel value

The CHANNEL parameter of 1 or greater value (the upper limit depends on the transducer block), chooses a data values from the transducer block. The channel value is visible in the SIMULATE parameter (transducer value and transducer status).

Simulation

This function is useful when starting up the plant. When a user switches SIMULATE.En/Disable to Enabled, the AI block start applying the simulation value and the simulation status as the channel value instead of the transducer value and the transducer status. This function becomes active only when the SIMULATE and simulation hardware jumper is turned ON. Do not forget to disable the SIMULATE and simulation hardware jumper after this function is used.

Recommended parameter values

Table 3.2 shows the three applications in the AI block and Table 3.3 shows the recommended values for those applications.

Note that majority of the parameters can be changed only when MODE_BLK.Target is in O/S mode.

Table 3.2 Three AI Block Applications

Application	Description
Application AI1: Simple measurement	The transducer block provides measured value in the desired unit. AI block can be transparent. In case of Yokogawa's EJA transmitter, its transducer block gives differential pressure in the unit given in XD_SCALE of AI block and this application can be applied in most cases.
Application AI2: Scaling in AI	The transducer block provides measured value in a unit and AI block needs to convert it into another unit. In case of a level meter with pressure transmitter, transducer block gives pressure and AI block converts it into level with a linear equation. Two set of values are needed: 0% Level = L0 inches = P0 psi 100% Level = L1 inches = P1 psi
Application AI3: Orifice plate flowmeter	The transducer block provides measured differential pressure with an orifice plate and AI block converts it into flow rate by calculating square-root of differential pressure. Two set of values are needed: 0% flow = F0 GPM = P0 psi 100% flow = F1 GPM = P1 psi

Table 3.3 Recommended Parameter Values of AI Block

Parameter Mnemonic	Application AI1	Application AI2	Application AI3
1. ST_REV	dynamic		
2. TAG_DESC	Any text		
3. STRATEGY	1		
4. ALERT_KEY	1		
5. MODE_BLK			
Target	AUTO		
Actual	dynamic		
Permitted	O/S+MAN+AUTO		
Normal	AUTO		
6. BLOCK_ERR	dynamic		
7. PV	dynamic		
8. OUT	dynamic		
9. SIMULATE			
Simulate Status	dynamic		
Simulate Value	dynamic		
Transducer Status	dynamic		
Transducer Value	dynamic		
En/Disable	Disable		
10. XD_SCALE			
EU@100%	100	P1	P1
EU@0%	0	P0	P0
Units Index	Transducer unit	psi	psi
Decimal Point	1	any	any
11. OUT_SCALE			
EU@100%	100	L1	F1
EU@0%	0	L0	F0
Units Index	Output unit	inch	GPM
Decimal Point	1	any	any
12. GRANT_DENY	dynamic		
13. IO_OPTS	Low Cutoff		
14. STATUS_OPTS	Propagate Fault Forward		
15. CHANNEL	Appropriate value starting from 1 to receive transducer block output		
16. L_TYPE	Direct	Indirect	Indirect Sq Root
17. LOW_CUT	0		
18. PV_FTIME	0		
19. FIELD_VAL	Dynamic		
20. UPDATE_EVT	Dynamic		
21. BLOCK_ALM	Dynamic		
22. ALARM_SUM	Dynamic		
Current	Dynamic		
Unacknowledged	Dynamic		
Unreported	Dynamic		
Disabled	0		
23. ACK_OPTION	Auto Ack Enabled (1)		
24. ALARM_HYS	0.5		
25. HI_HI_PRI	0		
26. HI_HI_LIM	+Infinity		
27. HI_PRI	0		
28. HI_LIM	+Infinity		
29. LO_PRI	0		
30. LO_LIM	-Infinity		
31. LO_LO_PRI	0		
32. LO_LO_LIM	-Infinity		
33. HI_HI_ALM	dynamic		
34. HI_ALM	dynamic		
35. LO_ALM	dynamic		
36. LO_LO_ALM	dynamic		

T030302E.ai

IMPORTANT

The numbers in front of the parameter names of the Table 3.3 show the index offset from the block header. The parameters (or components) in the grayed columns are often initialized on shipping to useless values by some vendors, to prevent a function block from being operational. Such parameters must be set to meaningful values as listed in the table.

3.3.2 AO block

The analog output (AO) block is designed to let users use a “standard” model of an output device such as a valve positioner. The AO block receives the control value from a control block and sends back the current control value so that the control block can calculate the next control value or track the current value if the AO block is not controlled by that control block.

The AO block holds bi-directional data flow. One (forward path) is the flow from control value input to a transducer block; the other (backward path) is the flow from the transducer block to the control value.

Figure 3.5 shows the AO block structure.

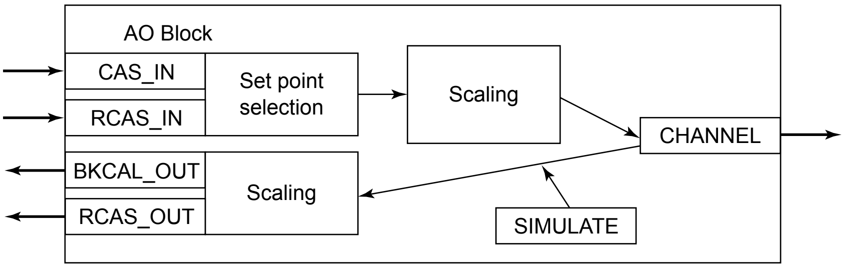


Figure 3.5 AO Block

■ Forward path

The control value from the controller becomes the setpoint (SP). An AO block has several paths to calculate SP, depending on the block mode. In CAS mode, CAS_IN subscribed from the publishing controller calculates the SP. In AUTO mode, SP value is provided by an on-demand Write request to SP. In RCAS (remote cascade) mode, remote controller provides data to RCAS_IN.

In CAS or RCAS mode, CAS_IN or RCAS_IN is scaled by PV_SCALE for the controller and then XD_SCALE scales it for the transducer block. In most cases those parameters have the same engineering units such as percent (%), and those scales of 0 and 100 simply correspond with EU@0% and EU@100%.

Before calculating SP, CAS_IN or RCAS_IN goes through limiting functions, upper/lower limits and ramp rate limit. For upper/lower limits, the value is replaced with PV_HI_LIM (PV_LO_LIM) if the value is higher (lower) than the limit. For ramp rate limit, the difference of current CAS_IN or RCAS_IN from the previous value is higher than SP_RATE_HI (or lower than SP_RATE_LO); another value with limited rate is adopted to calculate PV.

SP becomes OUT if the communication is going well. In case the controller disappears, either OUT does not change or it goes to the predefined fault-state and be replaced with FSTATE_VAL if options are appropriately selected.

OUT is given to the transducer block via CHANNEL.

Backward Path

An actuator's current position information is transmitted to the READBACK parameter (by the transducer block unit), converted into the specified range in the PV_SCALE before it is written to the PV parameter.

BKCAL_OUT is a parameter with a READBACK value to the PID block.

Usually, an SP value itself is transmitted to the BKCAL_OUT.

PV (current position) can be transferred to BKCAL_OUT; however, it is not used in the process control under ordinary condition.

Fault State

The fault state is provided for very critical processes where control valves need to be closed or opened when the upstream block is unavailable (broadcasting a BAD status) for a pre-determined time interval.

Modes

An AO block mode can be in O/S, MAN, LO (Local Override), AUTO, CAS (cascade), RCAS (Remote cascade), or ROUT (Remote output).

IMPORTANT

It is necessary to bring the AO block into CAS mode to set both CAS and AUTO bits in MODE_BLK. Target.

Recommended parameter values

Table 3.4 shows two of the applications in the AO block, and Table 3.5 shows the recommended parameter values for those applications. Note that majority of the parameters can be changed only when MODE_BLK.Target is in the O/S (out of service) state.

Table 3.4 AO Block Applications

Application	Description
Application AO1: Control with the current target position	AO block is used with PID in the FF function block. A tracking value from the AO block shows the current SP given by the PID or the manual operation.
Application AO2: Control with the current valve position	AO block is used with PID in the FF function block. A tracking value from the AO block shows PV, the current valve position.

Table 3.5 Recommended Parameter Values of AO Block

Parameter Mnemonic	Application AO1	Application AO2
1. ST_REV	dynamic	
2. TAG_DESC	Any test	
3. STRATEGY	1	
4. ALERT_KEY	1	
5. MODE_BLK		
Target	CAS+AUTO	
Actual	dynamic	
Permitted	O/S+MAN+AUTO+CAS+RCAS	
Normal	CAS+AUTO	
6. BLOCK_ERR	dynamic	
7. PV	dynamic	
8. SP	dynamic	
9. OUT	dynamic	
10. SIMULATE		
Simulate Status	dynamic	
Simulate Value	dynamic	
Transducer Status	dynamic	
Transducer Value	dynamic	
En/Disable	Disable	
11. PV_SCALE		
EU@100%	100	
EU@0%	0	
Units Index	%	
Decimal Point	1	
12. XD_SCALE		
EU@100%	100	
EU@0%	0	
Units Index	%	
Decimal Point	1	

Parameter Mnemonic	Application AO1	Application AO2
13. GRANT_DENY	dynamic	
14. IO_OPTS	SP-PV track in MAN +SP-PV track in LO(*1)	+Use PV for BKCAL_OUT
15. STATUS_OPTS	NONE	
16. READBACK	dynamic	
17. CAS_IN	dynamic	
18. SP_RATE_DN	+Infinity	
19. SP_RATE_UP	+Infinity	
20. SP_HI_LIM	100	
21. SP_LO_LIM	0	
22. CHANNEL(*1)	Use 1 for experiment	
23. FSTATE_TIME	0(*2)	
24. FSTATE_VAL	0(*2)	
25. BKCAL_OUT	Dynamic	
26. RCAS_IN	Dynamic	
27. SHED_OPT	Normal Shed Normal Return (1)	
28. RCAS_OUT	dynamic	
29. UPDATE_EVT	dynamic	
30. BLOCK_ALM	dynamic	

T030304E.ai

*1: Refer to the devices' instruction manuals or the device capabilities file (CF) for CHANNEL information.
*2: This configuration is designed for typical applications that a control valve has to maintain its current position when the communication status shows BAD due to the communication failure or block malfunction. In case of a very critical process, when the process data is unavailable, use the fault state option so that a control valve has to maintain the predefined safe position.
- IO_OPTS: "Fault State to value" + "Use Fault State value on start"
- FSTATE_VAL: Fault state of a control valve
- FSATE_TIME: A time (in second) from an occurrence of a failure to the commencing of the fault state action.

It is necessary to set "Fault State supported" to the FEATURE_SEL parameter in the resource block.

IMPORTANT

The numbers in front of the parameter names of the recommended parameter values show the index offset from the block header. The parameters (or components) in the grayed columns are often initialized on shipping to useless values by some vendors, to prevent a Function Block from being operational. Such parameters must be set to meaningful values as listed in the table.

3.3.3 PID block

A PID Block is a standardized model of a PID control function. The process value comes through the IN parameter and its control output is the OUT parameter. Other input/output parameters are provided for various control schemes such as cascade control. Figure 3.6 shows the PID block diagram.

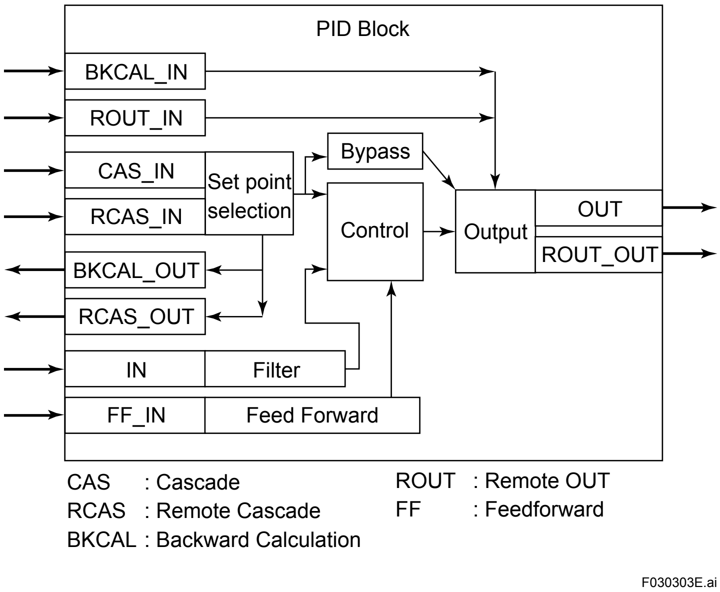


Figure 3.6 PID block

Modes

A PID block can be in various modes, O/S, MAN, IMan (initialization manual), LO (local override), AUTO, CAS (cascade), RCAS (remote cascade) and ROUT (remote output).

IMan (Initialization manual) mode

A PID block comes to IMan mode when the downstream AO block does not accept control from the PID (e.g., O/S, MAN or LO mode). The PID block tracks the current output of the downstream block such as AO block.

Normal mode

PID block is normally in AUTO (closed loop) or CAS (cascaded loop). O/S is used for to stop the PID block action. MAN can be used for output control.

Setpoint

A PID block receives control setpoints in several ways depending on its mode. The following two parameters are important.

SP

A setpoint can be directly written to the SP parameters when MODE_BLK.Target is AUTO, MAN, or O/S.

CAS_IN

In CAS mode, the PID block receives a setpoint via CAS_IN parameter. It comes from an upstream FF function block and the current setpoint is sent back via BKCAL_OUT parameter.

PID parameters

GAIN, RESET, and RATE are the tuning constants for the P, I, and D terms respectively. Gain is a number with no dimension. RESET and RATE are time constants expressed in seconds. They have another PID parameter, BAL_TIME, which can be set as zero (0).

Recommended parameter values

Table 3.6 shows two of applications in the PID block and Table 3.7 shows the recommended parameter values for those applications. Note that majority of the parameters can be changed only when MODE_BLK target is in O/S mode.

Table 3.6 PID block applications

Application	Description
Application PID1: Single loop	PID block provides OUT to AO or other PID blocks. SP to the PID block is provided by an operator.
Application PID2: Cascade loop	PID block receives OUT from other PID blocks.

Table 3.7 Recommended parameter values of PID block

Parameter Mnemonic	Application PID1	Application PID2
1. ST_REV	dynamic	
2. TAG_DESC	Any test	
3. STRATEGY	1	
4. ALERT_KEY	1	
5. MODE_BLK		
Target	AUTO	AUTO+CAS
Actual	dynamic	
Permitted	O/S+MAN+AUTO+CAS+RCAS+ROUT	
Normal	AUTO	AUTO+CAS
6. BLOCK_ERR	dynamic	
7. PV	dynamic	
8. SP	dynamic	
9. OUT	dynamic	
10. PV_SCALE		
EU@100%	100	
EU@0%	0	
Units Index	%	
Decimal Point	1	
11. OUT_SCALE		
EU@100%	100	
EU@0%	0	
Units Index	%	
Decimal Point	1	
12. GRANT_DENY	dynamic	
13. CONTROL_OPTS	No OUT limits in MAN +Obey SP limits if CAS or RCAS	+ 3 more (*1)
14. STATUS_OPTS	Target to MAN if BAD IN Target to next permitted mode if BAD_BAS_IN	
15. IN	dynamic	
16. PV_FTIME	0	
17. BYPASS	OFF (1)	
18. CAS_IN	dynamic	
19. SP_RATE_DN	+Infinity	
20. SP_RATE_UP	+Infinity	
21. SP_HI_LIM	100	
22. SP_LO_LIM	0	
23. GAIN	1	
24. RESET	10	
25. BAL_TIME	0	
26. RATE	0	
27. BKCAL_IN	dynamic	
28. OUT_HI_LIM	100	
29. OUT_LO_LIM	0	
30. BKCAL_HYS	0.5	

Parameter Mnemonic	Application PID1	Application PID2
31. BKCAL_OUT	dynamic	
32. RCAS_IN	dynamic	
33. ROUT_IN	dynamic	
34. SHED_OPT	Normal Shed Normal Return (1)	
35. RCAS_OUT	dynamic	
36. ROUT_OUT	dynamic	
37. TRK_SCALE		
EU@100%	100	
EU@0%	0	
Units Index	%	
Decimal Point	1	
38. TRK_IN_D	dynamic	
39. TRK_VAL	dynamic	
40. FF_VAL	dynamic	
41. FF_SCALE		
EU@100%	100	
EU@0%	0	
Units Index	%	
Decimal Point	1	
42. FF_GAIN	0	
43. UPDATE_EVT	dynamic	
44. BLOCK_ALM	dynamic	
45. ALARM_SUM	Other components are dynamic	
Disabled	0	
46. ACK_OPTION	Auto Ack Enabled (1)	
47. ALARM_HYS	0.5	
48. HI_HI_PRI	0	
49. HI_HI_LIM	+Infinity	
50. HI_PRI	0	
51. HI_LIM	+Infinity	
52. LO_PRI	0	
53. LO_LIM	-Infinity	
54. LO_LO_PRI	0	
55. LO_LO_LIM	-Infinity	
56. DV_HI_PRI	0	
57. DV_HI_LIM	+Infinity	
58. DV_LO_PRI	0	
59. DV_LO_LIM	-Infinity	
60. HI_HI_ALM	dynamic	
61. HI_ALM	dynamic	
62. LO_ALM	dynamic	
63. LO_LO_ALM	dynamic	
64. DV_HI_ALM	dynamic	
65. DV_LO_ALM	dynamic	

T030306E.ai

*1: For Application PID2, add one of the followings to the CONTROL_OPTS: [SP-PV tracks in MAN], [SP-PV tracks in ROUT], and [SP-PV tracking in LO or IMan].

IMPORTANT

The numbers in front of the parameter names of the recommended parameter values show the index offset from the block header. The parameters (or components) in the grayed columns are often initialized on shipping to useless values by some vendors, to prevent a Function Block from being operational. Such parameters must be set to meaningful values as listed in the table.

3.3.4 Resource block and transducer block

The mode of resource block and transducer block influences the FF function block behaviors and they must be set to AUTO for appropriate operation.

Various parameters of a transducer block are depending on its device functionality. The parameter definition and its behavior must be considered individually.

Table 3.8 and 3.9 show recommended values for the resource block and the transducer block. Note that majority of the parameters can be changed only when MODE_BLK Target is in O/S mode.

Table 3.8 Recommended parameter values of resource block

Parameter Mnemonic	Value
1. ST_REV	dynamic
2. TAG_DESC	Any text
3. STRATEGY	1
4. ALERT_KEY	1
5. MODE_BLK	
Target	AUTO
Actual	dynamic
Permitted	O/S+AUTO
Normal	AUTO
6. BLOCK_ERR	dynamic
7. RS_STATE	dynamic
8.TEST_RW	dynamic
9. DD_RESOURCE	fixed
10. MANUFAC_ID	fixed
11. DEV_TYPE	fixed
12. DEV_REV	fixed
13. DD_REV	fixed
14. GRANT_DENY	dynamic
15. HARD_TYPES	fixed
16. RESTART	Do not write here
17. FEATURES	fixed
18. FEATURE_SEL	Copy from FEATURES(*1)
19. CYCLE_TYPE	fixed
20. CYCLE_SEL	Copy from CYCLE_TYPE
21. MIN_CYCLE_T	fixed
22. MEMORY_SIZE	fixed
23. NV_CYCLE_T	fixed
24. FREE_SPACE	fixed
25. FREE_TIME	fixed
26. SHED_RCAS	640000
27. SHED_ROUT	640000
28. FAULT_STATE	dynamic
29. SET_FSTATE	OFF(1)
30. CLR_FSTATE	OFF(1)
31. MAX_NOTIFY	fixed
32. LIM_NOTIFY	Bigger one from (3, MAX_NOTIFY)
33. CONFIRM_TIME	640000
34. WRITE_LOCK	Unlocked(1)
35. UPDATE_EVT	dynamic
36. BLOCK_ALM	dynamic
37. ALARM_SUM	Other components are dynamic
Disabled	0
38. ACK_OPTION	Auto Ack Enabled(1)
39. WRITE_PRI	0
40. WRITE_ALM	dynamic
41. ITK_VER	fixed

T030307E.ai

*1: Do not set the fault state supported bit of FEATURES_SEL unless the fault state is required. Also refer to the note *2 of Table 3.5 AO block.

IMPORTANT

The numbers in front of the parameter names of the recommended parameter values show the index offset from the block header. The parameters (or components) in the grayed columns are often initialized on shipping to useless values by some vendors, to prevent a Function Block from being operational. Such parameters must be set to meaningful values as listed in the table.

Table 3.9 Reommended parameter values of transducer block

Parameter Mnemonic	Value
1. ST_REV	dynamic
2. TAG_DESC	Any text
3. STRATEGY	1
4. ALERT_KEY	1
5. MODE_BLK	
Target	AUTO
Actual	dynamic
Permitted	O/S + AUTO
Normal	AUTO
6. BLOCK_ERR	dynamic
7. UPDATE_EVT	dynamic
8. BLOCK_ALM	dynamic
9. TRANSDUCER_DIRECTORY	fixed
10. TRANSDUCER_TYPE	fixed
11. XD_ERROR	dynamic
12. COLLECTION_DIRECTORY	fixed
13. PRIMARY_VALUE_TYPE	fixed
14. PRIMARY_VALUE	dynamic
15. CAL_POINT_HI	Written by calibrator
16. CAL_POINT_LO	Written by calibrator
17. CAL_MIN_SPAN	fixed
18. CAL_UNIT	Written by calibrator
19. SENSOR_TYPE	fixed
20. SENSOR_RANGE	fixed
21. SENSOR_SN	fixed
22. SENSOR_CAL_METHOD	Written by calibrator
23. SENSOR_CAL_LOC	Written by calibrator
24. SENSOR_CAL_DATE	Written by calibrator
25. SENSOR_CAL_WHO	Written by calibrator
26. SENSOR_ISOLATOR_MTL	fixed
27. SENSOR_FILL_FLUID	fixed
28. SECONDARY_VALUE	dynamic
29. SECONDARY_VALUE_UNIT	

T030308E.ai

Note: This table shows typical transducer block parameters. Since the transducer block parameters are dependant on the physical principle, a user's transducer block may have different parameters. In any case, ensure that STRATEGY, ALERT_KEY and MODE_BLK are set to the above values.

IMPORTANT

The numbers in front of the parameter names of the recommended parameter values show the index offset from the block header. The parameters (or components) in the grayed columns are often initialized on shipping to useless values by some vendors, to prevent a Function Block from being operational. Such parameters must be set to meaningful values as listed in the table.

3.3.5 Unit Codes

The Fieldbus Foundation specified certain numeric codes for engineering units. Important engineering units are extracted here.

However, it is much easier to manipulate these engineering units by their “unit” text provided by DD rather than the numeric expressions.

Table 3.10 shows the list of engineering units and their numeric expressions for the convenience that the human machine interface requires only the raw data display. For other engineering units not listed here, refer to the documents available from the Fieldbus Foundation.

Table 3.10 Major unit codes defined by the FOUNDATION™ Technical Note 16 – Standard tables

pressure	
code	**unit**
1130	Pa
1132	MPa
1133	kPa
1134	mPa
1135	µPa
1136	hPa
1137	ba
1138	mbar
1139	torr
1140	atm
1141	psi
1142	psia
1143	psig
1144	g/cm^2
1145	kg/cm^2
1146	inH$_2$O
1147	inH$_2$O(4°C)
1149	mmH$_2$O
1150	mmH$_2$O(4°C)

mass	
code	**unit**
1088	kg
1089	g
1090	mg
1092	t
1093	oz
1094	lb

density	
code	**unit**
1097	kg/m^3
1100	g/cm^3
1102	t/m^3
1103	kg/l
1106	lb/in^3
1107	lb/ft^3

frequency	
code	**unit**
1077	Hz
1079	GHz
1080	MHz
1081	kHz
1082	1/s
1083	1/min
1085	RPM

flow	
code	**unit**
1322	kg/s
1330	lb/s
1351	l/s
1352	l/min
1353	l/h
1356	CFS
1357	CFM
1358	CFH
1363	GPM

volume	
code	**unit**
1034	m^3
1036	cm^3
1037	mm^3
1038	l
1039	cl
1040	ml
1517	kl
1042	in^3
1043	ft^3
1044	yd^3
1046	pint
1047	quart
1048	gallon
1050	bushel
1051	barrel

velocity	
code	**unit**
1061	m/s
1062	mm/s
1063	m/h
1064	km/h
1065	knot
1066	in/s
1067	ft/s
1068	yd/s
1069	in/min
1070	ft/min
1071	yd/min
1072	in/h
1073	ft/h
1074	yd/h
1075	MPH

temperature	
code	**unit**
1000	K
1001	°C
1002	°F

length	
code	**unit**
1010	m
1011	km
1012	cm
1013	mm
1018	feet
1019	inch
1020	yard
1021	mile

area	
code	**unit**
1023	m^2
1024	km^2
1025	cm^2
1027	mm^2
1030	in^2
1031	ft^2
1032	yd^2
1033	mile^2

electricity	
code	**unit**
1209	A
1211	mA
1234	V/m
1240	V
1242	kV
1243	mV
1281	Ω
1284	kΩ

time	
code	**unit**
1054	s
1057	µs
1058	min
1059	h
1060	d

miscellaneous	
code	**unit**
1342	%
1422	pH
1423	ppm
1424	ppb

T030309E.ai

3.4 System Management

For all the FOUNDATION fieldbus devices, system management is an important application which manages device information and its behavior in the FOUNDATION fieldbus system.

3.4.1 Device Management

A device connected to the FOUNDATION fieldbus is distinguished from other devices by one of three identifiers:

- Device identification (ID): A system independent identifier that is provided by the manufacturer. It is burnt into the device and never changed.
- Physical device (PD) tag: A unique system-specific name to identify a device used in the plant or a set of fieldbus segments assigned by the user.
- (Physical) node address: A 2-digit, hexadecimal number of eight-bit length assigned by a user through network configuration, and it is unique to a fieldbus segment.

A device ID is a non-volatile value in the device that is a globally unique identifier for the device. It is set by the device manufacturer and valid for device management purpose.

The PD tag is a system unique string that identifies the device by its function in the system. It is a 32-digit length text field. Upon replacing a failed device, it is common practice to give the same PD tag to the new device (as the old one).

Since the data length of the device ID and the PD tag are very long (32 bytes (*1)), it is not preferred to used them in the daily communications, especially in the 31.25 kbps low speed network. Instead, a node address is used for a device identifier in the communication, and services to correlate these three identifiers are provided.

For example, a pressure transmitter is shipped with a permanent device ID of "59454300031999DE C22001102344" and configured to have a PD tag of "FI1001" and a node address of 0xF5.

Field device's system management agent responds to the System Management Kernel Protocol (SMKP) requests from an administrator the device for device configuration. The functions of the system management agent are as folllows:

- Identify a device ID and a PD tag of device
- Sets/clears the node address
- Provides/clears a PD tag to a device, and
- Finds out a node address of a specific PD tag.

Note that even when a node address of a device is cleared, it must be able to join communication. For this purpose, a special address range (from 0xF8 to 0xFB) is reserved and a device without a node address can join the network using one in this address range.

*1: A physical device (PD) tag of 16 alphanumeric characters is used for Yokogawa's CENTUM production control systems.

3.4.2 FF function block management

As explained in Section 3.2.2 of this document, the FF function block algorithm must start at a defined time. The system management agent stores FF function block scheduling information and starts an assigned FF function block at the designated time.

Macrocycle is the overall application period and the schedule is designed as an offset time from the start of macrocycle.

3.4.3 Application time management

All the system management agents in a system track an application time (or system time). It is used to record an event with a time stamp.

The system time and link scheduling (LS) time are not identical. LS time is used locally in the data link layer for communication and FF function block execution in the system management. The system time is used in the application and is synchronized among all the devices in a system composed of multiple fieldbus segments.

3.5 Device support files

Applications such as for human-machine interface and fieldbus configuration need more device information. Several files are standardized by the Fieldbus Foundation for securing devices interoperability as well as to help engineers.

3.5.1 Device description

Device description (DD) provides information on block parameters. A function block parameter can be read by names and displayed properly according to its data type and display specifications.

This is very useful in handling enumerated parameters; for example, a user can use “kPa” instead of its code value 1133 without referring to an instruction manual. Whenever a new device is introduced, it becomes fully functional by simply installing the DD without updating the software in the host system.

The DD method can be run for a dedicated procedure (communication sequences) for calibration, diagnosis and others. The DD menu shows a list of DD methods and parameter sets.

The DD is useful for human-machine interface, system configuration and maintenance. Figure 3.7 shows an example of display using device description (DD).

The features of DD functions are as follows.

- Gives parameter information (especially non-standards parameters).
  - Name and descriptions
  - Data structure and attribute
  - Access right
  - Help menu
  - Calibration methods, etc.
  - Display format including graphical display, etc.
- Capable of displaying new devices with vendor-dedicated parameters on a standard screen.

Users are also able to perform ordinary process control and monitoring without the DD functions.

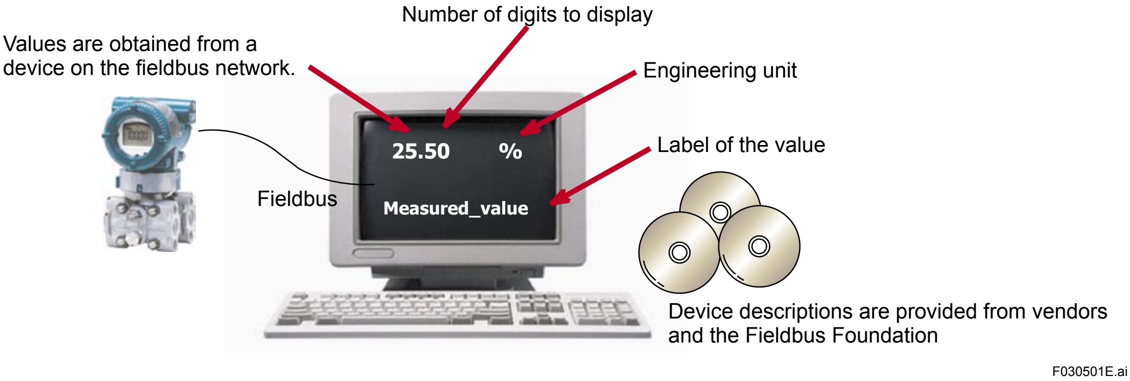


Figure 3.7 Example of a display using DD

What are DDL and DD?

DDL stands for “Device Description Language,” with which a device design engineer can describe the device functionality and data semantics. This text is then compiled by the “tokenizer” software to generate “DD binary” files.

The DD binary consists of two files: one is DD binary with the extension of “.ffo” and “.ff5,” and the other is DD symbol list with the extension of “.sym” and “.sy5.” Once the DD binaries are installed in users’ host systems, the users will gain full accesses to the function block parameters of the devices.

It is not practical for device vendors to individually write full information of the device, therefore, the

Fieldbus Foundation offers the DD Library, which provides a common DD and its dictionary. The device vendors have only to describe their dedicated part with DDL.

Figure 3.8 shows the DD’s hierarchical structure. The Fieldbus Foundation specifies the standard part of the blocks parameters and provides it as the DD “library.” Device “profiles” define a common part of various devices such as temperature transmitters and flowmeters. The vendor-specific part is specified in their DD files.

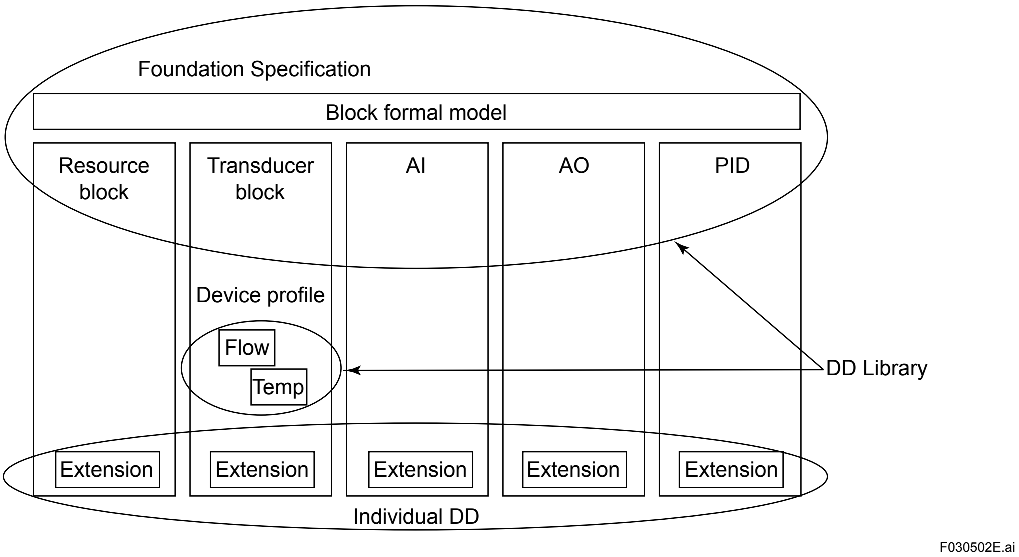


Figure 3.8 Hierarchical Structure of Device Description

Device description service

Device description service (DDS) is a software for a human-machine interface. It retrieves the information in the DD binary by using “DD item ID,” a key stored in the FMS object description of parameters.

The DD binary files are stored in the following directory structure:

<DD home directory>

+- Manufacturer ID

+--- Device type

Manufacturer ID is a unique code given from the Fieldbus Foundation to the manufacturers of field devices. Users can identify the device manufacturer by this ID. It is described in 3-byte length in a 6-digit hexadecimal expression. Yokogawa’s manufacturer ID is 594543 which represents “YEC.”

Device type is described in 2-byte length in a 4-digit hexadecimal expression. It distinguishes the target devices out of other devices by the same device manufacturer. The device type for Yokogawa’s EJA is 0003.

3.5.2 Capabilities file

The capabilities file contains all the information on the capability and constraints of a device to the users on the aspects of network/system management and FF function block.
A certain part of the information resides in the device itself; however, this file is useful for offline configuration of a fieldbus system without having real devices. A Capabilities file has an extension of ".cff." as it is often called "CFF" which means the common file format.

Figure 3.9 shows how the capabilities files are used during the system design and maintenance.

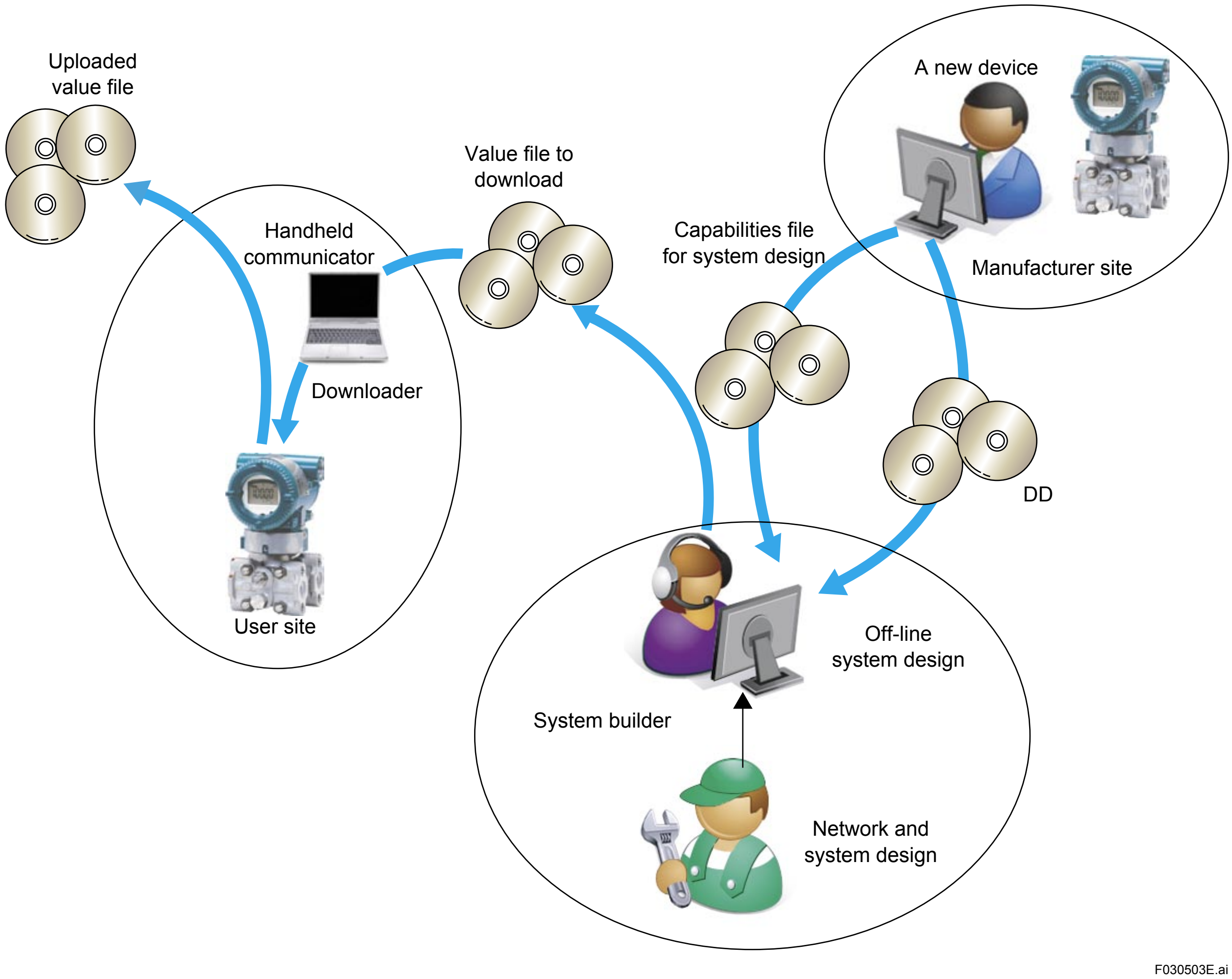


Figure 3.9 Use of Capabilities File

Blank Page

Revision Information

- Title : FOUNDATION Fieldbus Book – A Tutorial
- Manual No. : TI 38K02A01-01E

May 2001/1st Edition
Newly published
March 2003/2nd Edition

Title	Changed the title from “Fieldbus Book – A Tutorial” to “FOUNDATION Fieldbus Book – A Tutorial”
For all sections	Revised the section numbering and updated the contents
Introduction	Started with “Introduction”
	Added “Structure and Target Readership of This Textbook”
	Transferred the content of “Introduction” of 1st Edition into “Structure and Target Readership of This Textbook”
2.2.2	Revised the explanation about the electric equivalent circuit
2.2.3	Revised the description of wiring rules
3.2.1	Updated the list of Function Block (Table 3.1)
3.3.5	Corrected some unit codes
4	Revised the example project and the all descriptions related the example project
4.1.1	Added the consideration for the power supply capacity of Intrinsic Safe (IS) barrier in the Section “Power Supply and IS Barrier”
	Updated the description of the Section “Application Interaction”

Mar. 2012/3rd Edition
Revised and updated all pages.

Written by Yokogawa Electric Corporation

Published by Yokogawa Electric Corporation
2-9-32 Nakacho, Musashino-shi, Tokyo 180-8750, JAPAN