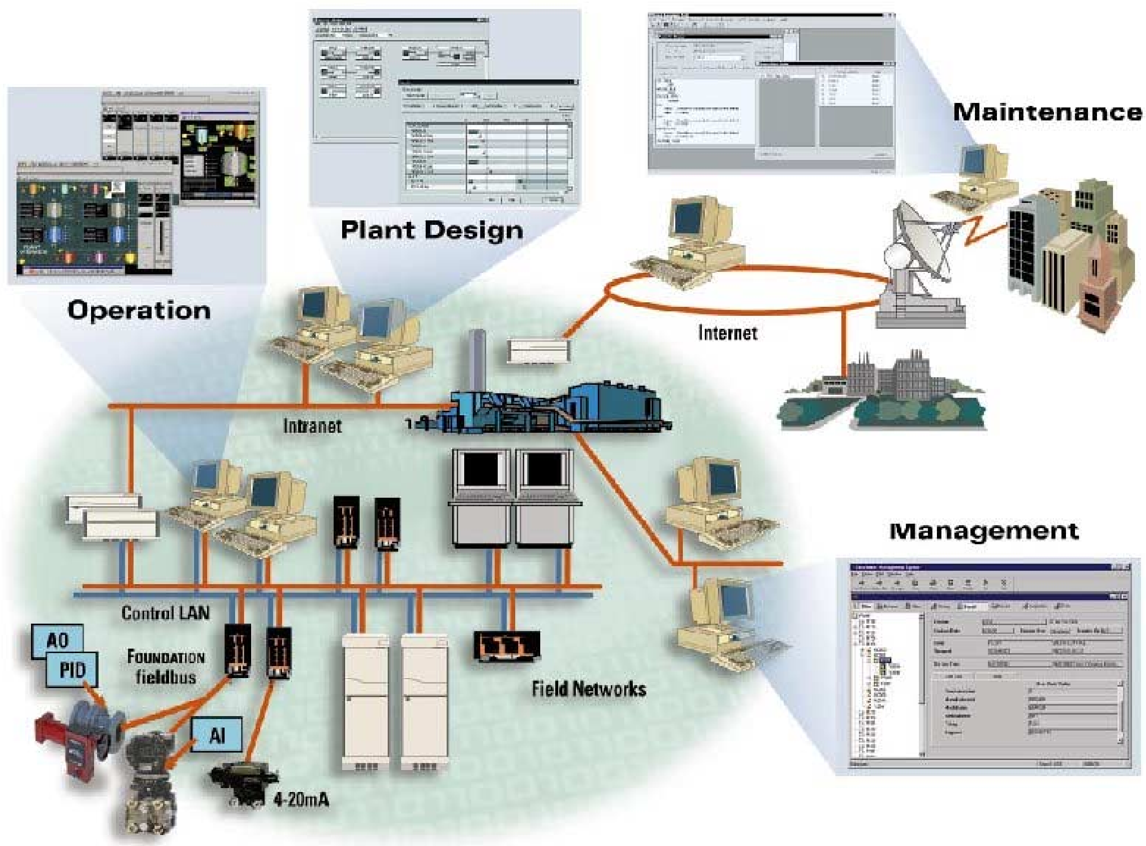
Technical Information

Fieldbus Book - A Tutorial

TI 38K02A01-01E



YOKOGAWA

Yokogawa Electric Corporation
2-9-32, Nakacho, Musashino-shi, Tokyo, 180-8750 Japan
Tel.: 81-422-52-5519 Fax.: 81-422-52-9802

TI 38K02A01-01E

1st Edition May 2001

Fieldbus Book - A Tutorial

This textbook is prepared for those who want to know more about technologies supporting FOUNDATION fieldbus and Function Blocks. Yokogawa Electric Corporation and its group funded this textbook to encourage FOUNDATION fieldbus adoption by as many people as possible. No commercial ambitions are in this book.

This textbook is a tutorial on fieldbus technologies and does not intend to override any technical content of FOUNDATION fieldbus. If this textbook contains any explanation that conflicts with Foundation documents, the Foundation documents are correct and such conflict should come from an error of explanation of this textbook or this textbook failed to follow the technology update of Fieldbus Foundation.

The authors want readers to use this textbook for getting more knowledgeable of FOUNDATION fieldbus and apply it to their industrial projects. **Fieldbus is the language of 21st century.**

Trademarks

Table of Contents

Introduction

This textbook is prepared for those who want to know more about technologies supporting FOUNDATION fieldbus and Function Blocks.

This textbook consists of four parts:

Introduction

Part A Communication Technologies

Part B Applications

Part C Managing Fieldbus Projects

If you have marketing interests in fieldbus, **Introduction** part gives you an overview. This part is, however, prepared for a quick overview before going to the details of technologies, and may not be enough for promotion activities. Refer to other documents to get more information.

Part A of this textbook gives you knowledge on communication technologies in FOUNDATION fieldbus. Implementation engineers and field engineers need a certain amount of communication technologies explained in this part.

Part B of this textbook explains applications running over fieldbus. They are very important for measurement and control on digital networks. All technical people working on digital instrumentation need such knowledge for real projects.

Part C of this textbook shows an example of fieldbus projects to make them successful. Project manager, instrumentation engineers and maintenance engineers are expected readers of this part.

What is Fieldbus?

Fieldbus Foundation defines "Fieldbus is 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.

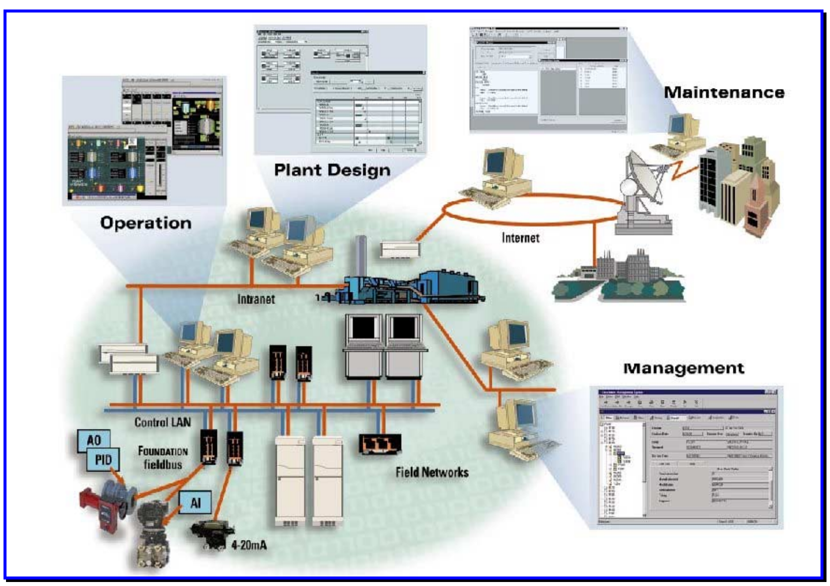
Modern industries could not survive without information technologies and networks in the 21st century. From production line to enterprise level, digital communication supports all economical and social activities by its latest and powerful technologies. Fieldbus is a part of it and cannot be separated from others. Fieldbus is the lowest level in the hierarchy and exchange information with higher-level databases.

We have "standards" to transfer measurement and control data between control room and plant floor. They are pneumatic and 4-20mA electric signals. We have enjoyed their finest features of interoperability and easy maintenance. Smart communication introduced in mid 80's opened an era of digital communication, but it had many limitations such as proprietary protocols, slow transmission speed and different data formats.

The idea of fieldbus was proposed to solve such problems. A "standardized" digital communication for industrial automation is changing the production system very quickly.

Fieldbus Benefits

Fieldbus is expected to reduce the life-cycle cost of production line and then total cost of ownership (TCO) of the plant.



Planning Phase

Fieldbus allows you to integrate your plant assets on a single plant automation system on digital communication networks. You can connect devices from multiple suppliers without custom software and these network-based systems allow for smaller control rooms, smaller cabinet rooms and more information productivity.

Installation Phase

Fieldbus offers reduced installation and material cost by replacing the traditional one-to-one wiring scheme with networking or multi-drop configuration, while intelligent field instruments make commissioning and plant startup much faster and less expensive.

Operation Phase

Fieldbus integrates various installations of control functions into one system to effectively optimize control of your plant. In addition, a unified human-machine interface (HMI) is provided for your operation. Function Blocks allow control functions to migrate into field devices allowing control functions to move to the field.

Maintenance Phase

Fieldbus allows for the reporting of self-diagnostics, calibration, and environmental conditions of field instruments without disturbing the plant control. Since it uses intelligent instruments, your stock for spare or replacement instruments can be dramatically reduced. Software packages for asset management are useful to minimize maintenance costs.

Renovation Phase

Enhanced functionality of field instruments is endless. Fieldbus devices are becoming standard off-the-shelf instruments, which make it very cost-effective and easy for you to extend the life of your plant. By simply connecting a new device, you can immediately benefit from advanced functionality. And, upgrade costs can be reduced because network-based systems are modular, which means they are done on-line.

FOUNDATION Fieldbus

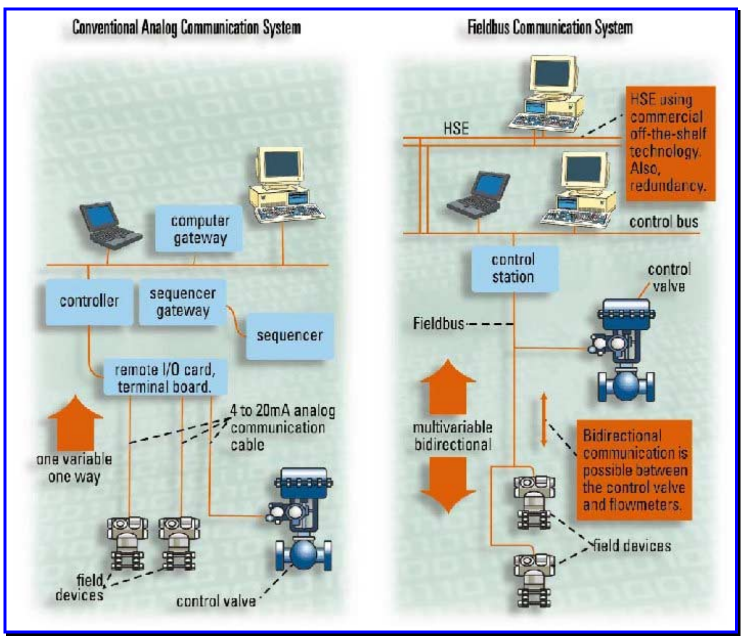
Fieldbus is not a product but a technology to make above benefits available to end-users. The following two conditions are necessary to make them come true:

- Many vendors provide fieldbus instruments.
- Those devices are interoperable.

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,
- To deliver FOUNDATION fieldbus specification,
- To provide technologies for fieldbus implementation including education, and
- To install an infrastructure to achieve interoperability.

FOUNDATION fieldbus is a subset of IEC/ISA standard (IEC61158 and ISA s50.02). Fieldbus Foundation and its members adopt FOUNDATION fieldbus as an enabling technology to utilize it to bring the above benefits to end-users.



Part A Communication Technologies

This part explains fundamental communication technologies to support Function Blocks and other applications. Though communication technologies are not visible to users except wiring, certain knowledge of underlying mechanism is often helpful to understand how Function Blocks serve your plant. If you have knowledge on fieldbus communications, or you want to know Function Blocks quickly, you may skip this part and go to Part B of this textbook.

This part gives you the knowledge on how fieldbus operates and supports your applications. You will understand how carefully fieldbus is designed for applications of industrial automation. Technology described here is FOUNDATION fieldbus, which is the specification of Fieldbus Foundation and a subset of the IEC/ISA international standard.

1. Communication Models

1.1 OSI Reference Model

Communication specifications are often explained with reference to the Open System Interconnect (OSI) layered model. 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 A-1 for OSI model. Layers 2 to 7 are implemented mostly by software and therefore are often called the “communication stack.”

Fieldbus Foundation specifies not only communication but also some user applications, which use FOUNDATION fieldbus communication, though the OSI model does not specify any user application.

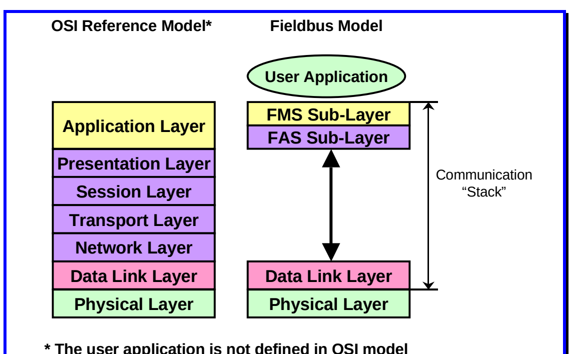


Figure A-1

Application Layer of Foundation fieldbus consists of two sublayers: Fieldbus Access sublayer (FAS) and Fieldbus Message Specification (FMS). FAS is the “glue” to map FMS services to Data Link Layer.

Figure A-2 shows the architecture of FOUNDATION fieldbus.

1.2 Protocol Data Unit

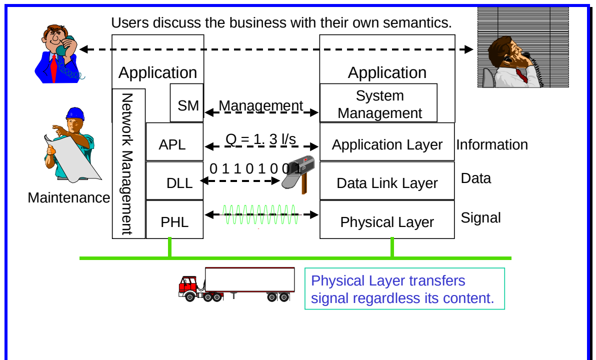


Figure A-2

Figure A-3 shows how a user data is transferred over the FOUNDATION fieldbus. Each layer appends layer-control information called Protocol Control Information (PCI) and more information to the message of the higher layer.

A data unit exchanged between the same layers is called “Protocol Data Unit (PDU).” A PDU may contain an optional data called “Service Data Unit (SDU),” which is a PDU of the next higher layer. A communication layer exchanges other PDUs without SDU to perform its functionality.

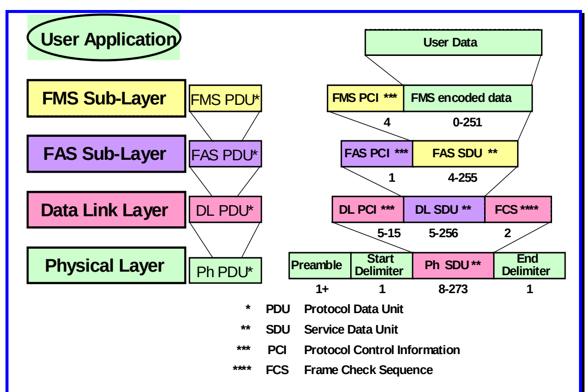


Figure A-3

1.3 Communication through VCR

Messages are exchanged between applications sitting on the FOUNDATION fieldbus. When a message is transferred, it goes down through a channel called Virtual Communication Relationship (VCR) to add PCI before it goes to the wire. At the destination, it goes up through the partner VCR to the receiving application. PCIs are appended and removed when a message goes through VCRs to allow layers to perform their specific functionality. See Figure A-4.

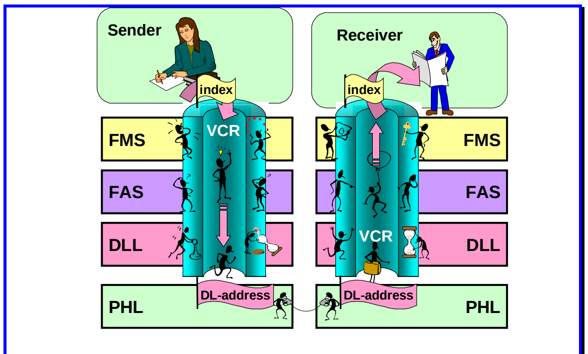


Figure A-4

A fieldbus device has many VCRs so that it can communicate with various devices or applications at the same time. It is possible because the VCR guarantees the message goes to the correct partner without risks of losing information. A VCR is identified by an application with device-local identifier called "index" specified in Application Layer. It is also identified from other devices with DL-address specified in Data Link Layer. A VCR has a queue (fast-in, fast-out memory) or a buffer (memory to store data) to save messages.

It is the responsibility of network configuration to give the correct information of the index and DL-address as well as other operating information to VCRs through Network Management.

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. Physical Layer interests are wires, signals, waveform, voltage, and others all related to electricity and optics.

Though the IEC/ISA standard specifies various media with various speeds, Fieldbus Foundation chose its own subset, low speed wire and fiber media, and Ethernet. Ethernet is discussed in a later chapter of this textbook.

2.1 31.25kbps Physical Layer

31.25kbps Physical Layer is the most common since IEC and ISA approved it in 1992. This textbook explains only wire medium but optical fiber can be used as an alternative.

Though 31.25kbps sounds slow compared with the latest telecommunication technologies, it is necessary to replace traditional 4-20mA analog transmissions. It is intended to apply to field devices for various environments of industries. Many users want to enjoy the simple installation of two-wired transmitters. Explosive gases prevent high-performance electronics in the hazardous area of the plant. Transmission of 31.25kbps was chosen for those applications, which demand devices of very low power consumption.

A field device can draw electric current from the medium to feed energy to its electronics. It is called a "bus-powered" device and is the fieldbus-equivalent of two-wire installation.

2.2 Signaling Method

Fieldbus utilizes a similar technology for smart transmitters to transmit an electric signal to the wire. Figure A-5 shows the electric equivalent circuit of signal transmission. Supply voltage is applied by a power supply through an impedance conditioner, typically consisting of inductors. DC current through the impedance conditioner feeds devices. Supply voltage is between 9V and 32V at the device terminals. The impedance conditioner makes output impedance of the power supply higher than 400Ω in the signal frequency bandwidth.

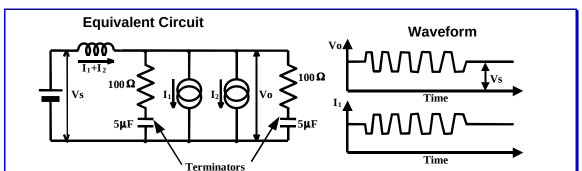


Figure A-5

Each cable end is terminated with a terminator of 100Ω impedance. It makes an instrumentation cable a balanced transmission line so that a signal of relatively high frequency can be transmitted with a minimum distortion.

When a device increases its sink current by 10mA, it is fed from capacitors in terminators because the impedance conditioner of power supply prevents current change through inductors. Thus the voltage between the wire pair decreases with 0.5V (= 10mA × 50Ω). In the next moment, the device increases drawing current by 20mA to generate a modulated signal of 1V p-p amplitude while the average current remains constant.

Data is encoded as a voltage change in the middle of one bit time. Data one (1) is encoded as a voltage fall in the middle of the bit time, while zero (0) is encoded as a voltage rise. Additional out-of-band data are N+ and N- encoded as constant voltage during the bit time. They are used only for start and stop delimiters to encode the start and end of PHL SDU (= DL PDU) so that Physical Layer can transmit any combinations of zeros and ones in DL PDU.

Figure A-6 shows the typical waveform of a Physical Layer signal. The receiving Physical Layer retrieves bit time using the preamble and then the boundary of octets (bytes) using the start delimiter. The end delimiter indicates the end of the Physical Layer signal. Preamble length can be increased when the signal goes over repeaters.

2.3 Wiring Rules

The IEC/ISA standard specifies minimum amplitude and worst waveform of a received signal at a device at the any place of the fieldbus network. The Physical Layer receiver circuit must be able to receive this signal.

You can configure the transmission line in any way as long as the received signal quality is guaranteed at all receiving nodes. However, it is not always simple and easy. Instead, ISA SP50 committee created a set of wiring rules to simplify the network design. The received signal is always of better quality than the minimum requirement if the fieldbus is designed according to these rules. They sound a little conservative but are useful to design a workable network very easily.

Rule 1: Number of devices on a fieldbus is between 2 and 32.

Rule 2: Cable is a twisted pair of individual shield (type A) with 18 AWG wires.

Rule 3: Overall cable length does not exceed 1900m.

Rule 4: Overall spur length does not exceed 120m. Maximum number of devices decreases according to the spur length as shown in the Table A-1.

Rule 5: When you use multiple twisted pairs with overall shield (type B) with 22 AWG wires, the total length decreases to 1200m.

Total Cable Length	Cable
(Type A) 1900m	Twisted pairs with individual shield
(Type B) 1200m	Twisted pairs with overall shield

Table A-1 (a) Total cable length (Rule 3, 5)

Devices on the fieldbus	Total Spur Length
1-12	120m
13-14	90m
15-18	60m
19-24	30m
25-32	1m

Table A-1 (b) Spur length (Rule 4)

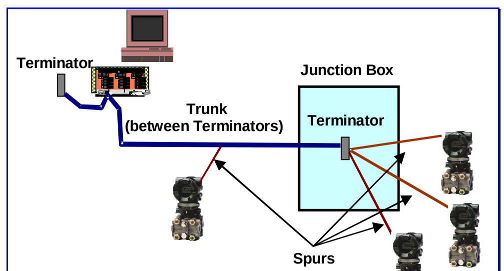


Figure A-7

Other cable types are defined but not recommended.

You can check whether above rules are met when you see the cable installation drawing. Note this is not the only rule to limit the number of devices on a fieldbus. Other rules are discussed later.

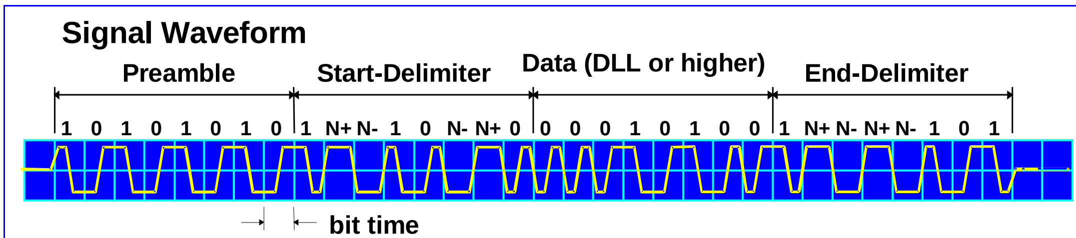


Figure A-6

2.4 I.S Consideration

Intrinsic Safe (I.S.) installation is important for plants where explosive gases exist. I.S. is the rule 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 I.S. field device must be carefully designed to prevent ignition even when a single failure of its component takes place.

An I.S. barrier must be installed to separate the hazardous area from the safe area as shown in Figure A-8. A barrier strictly limits the voltage, current and power fed to a device installed in the hazardous area. Therefore a field device must be operational with the restricted power supply. Devices and barriers must meet the same design criteria provided by safety organizations (IEC, FM, CENELEC, PTB, etc.).

Note that I.S. is the only possible technology for zone 0 where explosive gases exist at any time. In zone 1 where such gases exist most of the time but not always, explosion proof technology is also applicable as well as I.S. Explosion proof is a technology of housing design and is independent from fieldbus technologies.

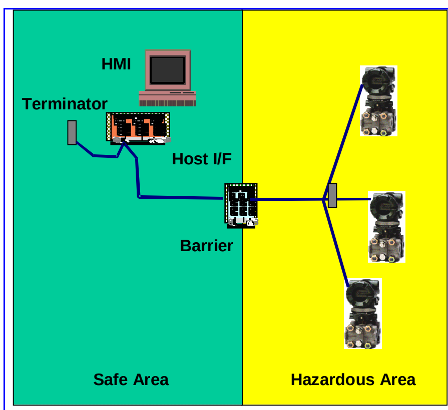


Figure A-8

3. Data Link Layer

Data Link Layer is a mechanism to transfer data from a node to the other nodes that need the data. It also manages the priority and order of such transfer requests. Data Link Layer interests are data, address, priority, medium control, and others all related to message transfer. Since Data Link Layer operates on the low speed Physical Layer, it has mechanisms to use the medium in an effective way.

Foundation fieldbus Data Link Layer is a subset of ISA s50.02 part - 3/4 and type 1 in IEC61158-3/4.

3.1 Medium Access Control

The most important functionality of Data Link Layer is Medium Access Control (MAC) of the fieldbus. Since all devices on the same cable receive the same Physical Layer signal, only one of them is allowed to transmit signal at a time. MAC is the method to achieve this goal. The domain of devices sharing the same Physical Layer signal is called a "link." In other words, only one device on a link is allowed to use the medium (Physical Layer) at a time.

Link Active Scheduler (LAS) has the role to control the medium access. Its functionality is explained in the following sections. The right to send a PDU is called a "token." The LAS possesses the token and gives it to another device to allow it to send messages. The token is then returned to the LAS for further medium access control.

Since application messages have various levels of urgency, Data Link Layer supports a mechanism to transmit messages according to their urgency. Data Link Layer provides three levels of "priority," URGENT, NORMAL and TIME_AVAILABLE, in this order. An URGENT message is transmitted immediately even when other messages of NORMAL or TIME_AVAILABLE priority are in the waiting queue, and vice versa. Maximum data size allowed for each priority is shown in Table A-2.

Priority	Maximum Data (DLSDU) Size
URGENT	64 bytes
NORMAL	128 bytes
TIME_AVAILABLE	256 bytes

Table A-2 Maximum Data Size in each Priority

3.2 Addresses

Communication partners in Data Link Layer are identified with DL-address, which consists of three components, *Link*, *Node* and *Selector*. See Figure A-9. *Link* field consists of 16 bits and identifies a "link." When the communication is within a link, this field is often omitted. This field is necessary when a message is going to other links through bridges.

Link (16 bit)	*Node* (8 bit)	*Selector* (8 bit)

Figure A-9

Node field gives the node address of 8 bits. A Foundation device has a node address in the ranges between 0x10 and 0xFF, which is classified into LM range, BASIC range, default range and temporary range. Usually devices are in LM or BASIC range according to its device class. When a device loses the node address, it communicates using one address in the default range. A temporary device such as a handheld communicator has node address in the temporary range. Link Active Scheduler has a node address of 0x04.

Figure A-10 shows the address range used in a fieldbus link. There is an address gap of size V(NUN). If a device has an address in this gap, it will never join the link. V(FUN) and V(NUN) are parameters you can access through Network management.

0x10 ~ V(FUN)	Address for Link Master class devices
V(FUN)+V(NUN) ~ 0xF7	Address for Basic class devices
0xF8 ~ 0xFC	Default address for devices with cleared address
0xFD ~ 0xFF	Address for temporary devices like a handheld communicator

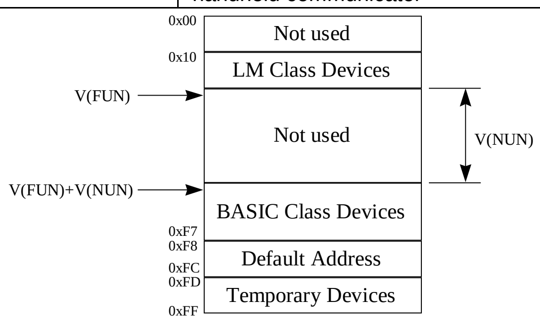


Figure A-10

Selector field gives a device-internal address of 8 bits to identify a VCR. When a VCR is connected to another VCR, it is identified with DLCEP (Data Link Connection End Point) shown in this field. When a VCR is not connected to any others but open to send/receive messages, it is identified with DLSAP (Data Link Service Access Point) shown in this field. DLCEP and DLSAP have different ranges.

Several DL addresses are reserved for specific purposes. For example, devices can share the same “global” DLSAP for alarm reception.

3.3 Link Active Scheduler

Link Active Scheduler (LAS) has the role to control the medium access.

Foundation devices are classified with device classes: 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. 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 A-11 shows the procedure through which a Link Master class device becomes the LAS.

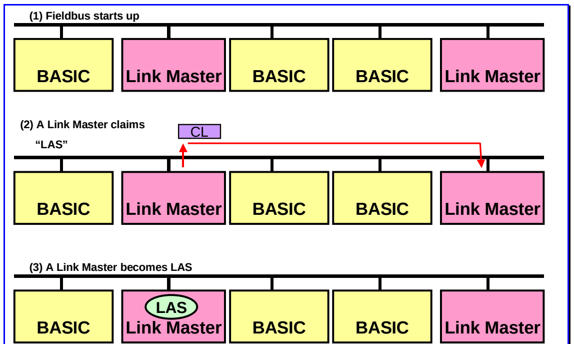


Figure A-11

Note that the LAS is an additional functionality to basic communication. Therefore it has a different DL-address (0x04) than the node address.

3.4 Scheduled Communication

The LAS is responsible for scheduled communication, which is necessary to link Function Blocks. Function Blocks are distributed applications operating in a synchronized manner. The LAS manages the communication part of the synchronized data transfer.

A Function Block output parameter is a “Publisher” of data and other Function Blocks that receive this data are called “Subscribers.” The LAS controls periodic data transfer from a Publisher to Subscribers using the Network Schedule.

When the time of scheduled communication comes, LAS sends Compel Data (CD) PDU to the Publisher DLCEP. Publisher is expected to transmit Data Transfer (DT) PDU stored in the data buffer of the DLCEP immediately. When Subscribers receive CD to the Publisher, they presume the next data transfer comes from the Publisher. Received data is stored in the buffer of Subscribers. A CD PDU is a token for a Publisher and the LAS interprets the publishing DT PDU as the returned token. See Figure A-12 (A).

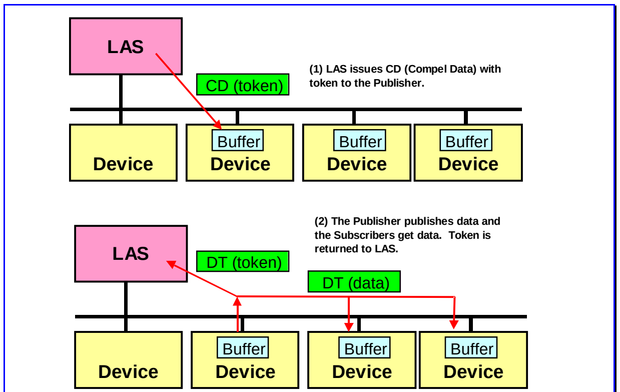


Figure A-12 (A)

Data Link Layer appends “freshness” information as PCI to the data so that the Subscribers know whether data has been updated since the last publish.

3.5 Unscheduled Communication

Other communications take place in an asynchronous way. The LAS is responsible to give all nodes on a link a chance to send messages. The LAS gives a token by sending Pass Token (PT) PDU to a node. A PT PDU contains priority and time interval information. 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 token as Return Token (RT) PDU.

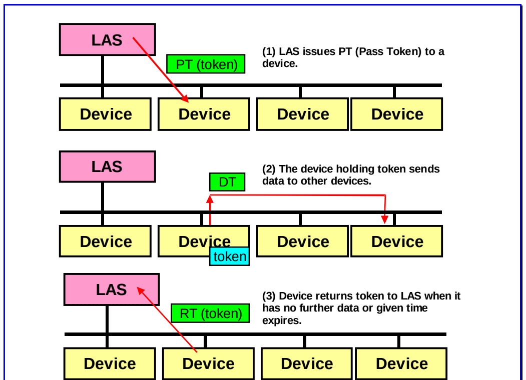


Figure A-12 (B)

The LAS controls the message transfer by updating the priority. When the token is given to all devices in a short time interval, the LAS gives more time to the nodes by lowering the priority. When the token does not go to all devices within a “target token rotation time” network parameter, the LAS increases the priority so that the token is given to all devices in a desired time interval.

A device must return the token within the time interval given in the PT PDU. This is necessary to finish the unscheduled communication before the next scheduled communication.

Note that the token is given to the node instead of DLCEP or DLSAP. Therefore the device is responsible to allow all DLCEPs and DLSAPs in the device to send messages.

3.6 Link Maintenance

The third role of LAS is to maintain the link. The LAS gives the token to all devices detected by the LAS. When a new device is added to the network, it must be recognized by the LAS and entered to the token rotation list called “Live List”.

The LAS sends a Probe Node (PN) PDU to node addresses where a device was not found before. A new device waits until it receives PN and returns Probe Response (PR) PDU to the LAS. 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 textbook. This probing is repeated in a given interval.

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

Whenever a change is detected in the Live List, the LAS broadcasts the change so that all LM devices share the latest list and are ready to take over.

The LAS also broadcasts its Data Link Time (LS-time) to the link in a predefined interval so that all devices on the network share the same time, which is necessary to start Function Blocks. It is often called “network time.”

3.7 Data Link PDUs

Table A-3 summarizes Data Link Protocol Data Units (DL PDUs) in FOUNDATION fieldbus.

DLPDU	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 CT
RQ	Round-trip Time Query	Measure delay in CT
RR	Round-trip Time Response	
CL	Claim LAS	Becomes LAS
TL	Transfer LAS	Request LAS role
IDLE	Idle	No activity

Table A-3 - DL PDUs

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.

4.1 Fieldbus Access Sublayer

Fieldbus Access Sublayer (FAS) is a part of secure communication. Since fieldbus does not have layers (4 to 6) between DLL and APL, FAS directly maps APL requests to DLL services. This is the most important part of VCR management.

The FAS provides three communication models for applications. They are explained below and summarized in Table A-4. It is expected a network manager configures VCRs correctly according to the models communicating with each other. Once configured, FAS provides the communication facility according to these models.

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

Table A-4 – Communication Models in FAS

4.1.1 Client-Server Model

The Client-Server model is universal and used in many communication technologies. An application called “Client” requests another application called “Server” to do a specific action through FMS. When the Server finishes the requested action, its result is transferred to the Client. It is a one-to-one two-way communication using DLCEP.

A typical example is a human-machine interface (Client) to read data of a Function Block (Server). The Client sends a Read request to the Server and then Server sends back the data to the Client. This communication takes place at any moment.

A Client may want to issue many requests at a time. Client-Server VCR has a queue to store those requests and sends requests one by one when the node has a token. A flow-control mechanism is available to manage error recovery and Server’s processing power.

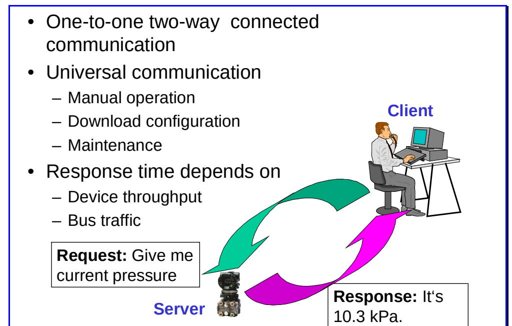


Figure A-13

4.1.2 Publisher-Subscriber Model

Publisher-Subscriber model is designed to link Function Blocks. When a publishing Function Block runs, its output data is stored in the buffer of the Publisher VCR. Then the LAS sends CD to the VCR to force it to transfer the data in DT PDU. Subscriber VCRs receive this PDU and gives the data to the subscribing Function Blocks.

A typical example is a linkage from output of an Analog Input (AI) block to process value input of PID control block.

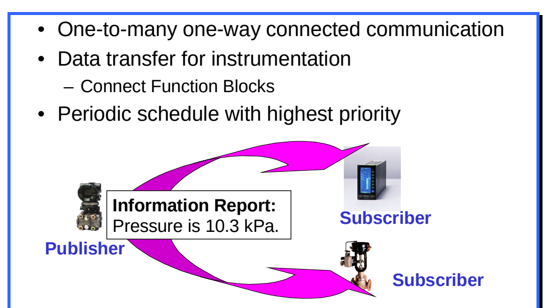


Figure A-14

The Publisher-Subscriber model is one-to-many one-way communication using DLCEP. Subscribers are able to know whether data has been updated since the last publish. This mechanism is important because Data Link Layer transfers data as scheduled regardless if the publishing Function Block updates the data in the buffer.

4.1.3 Source-Sink Model

The Source-Sink model is designed to broadcast messages. It is one-to-many one-way communication without schedule. This model is sometimes called "Report Distribution Model."

A Source VCR transfers a message in the queue to an assigned global DLSAP address when the device has the token. Sink VCRs have the same global address and receive the same message from a Source.

Foundation devices use this model for two specific purposes. One is to report alarms or events detected in the Source, and the other is to transmit trend of Source Function Block. Alarms are acknowledged through a Client-Server VCR.

It is desirable for an alarm logger to receive alarms from devices with one VCR. A Sink can receive messages from many Sources if the Sources are configured to send messages to the same global address. A Sink can identify the Source with its DLSAP address.

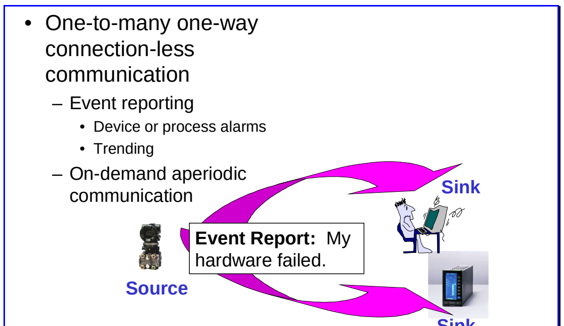


Figure A-15

4.2 Fieldbus Message Specification

Fieldbus Message Specification (FMS) is a service interface for user applications to use fieldbus services. When a service is requested, it encodes the request to transfer it to the other applications. The receiving FMS decodes the request to notify the application.

4.2.1 Virtual Field Device (VFD)

A fieldbus device may have user applications, which are independent from each other and do not interact. A fieldbus device consists of Virtual Field Devices for such individual applications. An identifier, given to a VCR, identifies the VFD.

A Foundation device has at least two VFDs. One is Management VFD where network and system management applications reside. It is used to configure network parameters including VCRs as well as to manage devices in a fieldbus system.

The other is a Function Block VFD where Function Blocks exist. It is possible for a field device to have two or more Function Block VFDs.

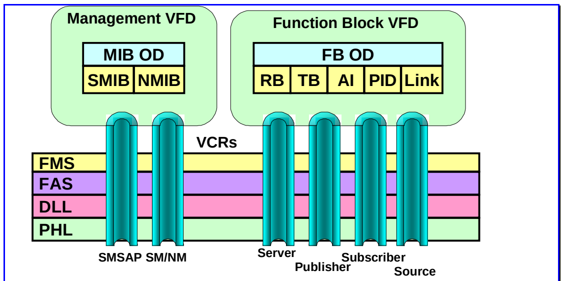


Figure A-16

4.2.2 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 access methods.

4.2.2.1 Object Examples

Function Blocks have parameter objects to which another application can have access. Alarms, Function Block Linkage are also objects. Their behavior is specified in the Function Block Application specification.

Network behavior is managed through Network Management Information Base (NMIB) objects. System behavior is managed through System Management Information Base (SMIB) objects. Schedules and VCRs are also objects.

4.2.2.2 Object Dictionary

An object is identified with a number called "index," which is unique within the VFD. Additional information to describe an object is necessary for open systems. Such information is called the "Object Dictionary (OD)," which is an assembly of information called "Object Descriptions" to explain the objects.

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

The most fundamental object is a "variable" to contain a value. It may be a simple variable, a record (structure) or an array. Function Block parameters, VCR, NMIB and SMIB are examples of record variables.

Other objects are event, domain, and program. They are explained in the next sections.

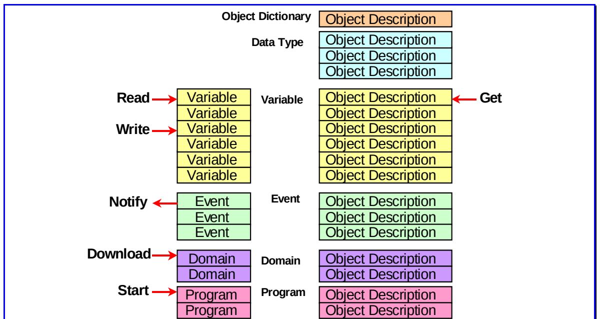


Figure A-17

An object is accompanied with its Object Description sharing the same index. There are Object Descriptions without associated objects. Those Object Descriptions give other information such as object location, amount of objects, data type, data structure and so on.

4.2.3 FMS Services

FMS provides services to access FMS objects. Tables summarize them with service classes.

4.2.3.1 Variable Access

A variable is storage of data. Its value can be read or written by another application. An application can send variable data without request from another application using *Information Report* service. Foundation fieldbus uses this service in publishing data and reporting trends. It is possible to define a list of variables for an effective transfer.

When a variable is a record or an array and consists of multiple variables, it is possible to transfer it as a whole or only one component assigned with "sub index."

Read	Read value of a variable
Write	Write value to a variable
Information Report	Send value as Publisher or Source
Define Variable List	Define a list of variables to send
Delete Variable List	Delete a list of variables

Table A-5 – Variable Access Services

4.2.3.2 Event Management

Event is used to notify that an application detects something important. Failure, data update and alarms are examples of events. An event is notified with the Source-Sink model repeatedly until it is acknowledged through the Client-Server model. Its notification can be enabled or disabled through another Event-related service.

Event Notification	Report an event as Source
Acknowledge Event Notification	Acknowledge an event
Alter Event Condition Monitoring	Disable or enable an event

Table A-6 – Event Services

4.2.3.3 Domain Management

Domain is a continuous memory area. It may be a program area or data area. A Client can download data to a domain or upload domain content through FMS services.

Because a domain can be larger than the maximum size of FMS encoding, FMS allows to upload or download a domain in parts. *Initiate* and *Terminate* services are prepared to manage partial download and upload.

Request Domain Download	Request download
(Generic) Initiate Download Sequence	Start downloading
(Generic) Download Segment	Download
(Generic) Terminate Download Sequence	Stop downloading
Request Domain Upload	Request upload
Initiate Upload Sequence	Start uploading
Upload Segment	Upload
Terminate Upload Sequence	Stop uploading

Table A-7 – Domain Services

4.2.3.4 Program Invocation

Program is a data processing functionality that can be managed from other applications. It was modeled for PLC ladder programs and can be used for Function Block Applications.

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

Create Program Invocation	Create a Program object
Delete Program Invocation	Delete a Program object
Start	Start a program
Stop	Stop a program
Resume	Resume a program execution
Reset	Reset the program
Kill	Disable the program

Table A-8 – Program Services

4.2.3.5 Other Services

FMS provides other services for Object Dictionary and Context management.

An Object Description can be read by the "*Get OD*" service. When an object is downloadable, its object descriptions need to be downloaded too.

A connection between applications is managed though Context. *Initiate* and *Abort* services are fundamental and the status information of the partner can be transferred by other services.

OD Management Services	
Get OD	Read an object description
Initiate Put OD	Start downloading OD
Put OD	Download an OD
Terminate Put OD	Stop downloading OD
Context Management Services	
Initiate	Establish a FMS connection
Abort	Release a FMS connection
Reject	Reject an improper request
Identify	Ask VFD Identification (vendor, model)

Table A-9 – Other FMS Services

5. System Management Protocol

System Management needs additional protocols to manage fieldbus systems. It must be operational even under abnormal situations such as system startup, wrong configuration, device failure and its replacement. Its protocol is called "System Management Kernel Protocol (SMKP)" and it directly uses Data Link Layer services without the Application Layer.

5.1 Tag and Address Assignment

A field device is identified with its PD tag as well as its node address as explained in the Application part of this textbook. SMKP provide services to assign them to a device.

When a device changes it address, it disappears and comes back to the link with a different address. Therefore its behavior is complicated in the Data Link Layer. A special protocol is defined for this purpose.

5.2 Tag Location

Device Tag and Function Block Tag are useful for humans but need longer data for communication. SMKP provides services to replace Device Tag and Block Tag with node address and index to make further communications much simpler.

5.3 Application Time Synchronization

Fieldbus Applications need to be synchronized in the sense of time to interact each other. For example, an event message needs a time stamp to indicate when it is detected, because it is received sometime later depending on the token rotation and bus traffic. SMKP provides a mechanism for all Management VFDs to share the synchronized time.

6. High Speed Ethernet

Though the Fieldbus specification allows faster media such as 1Mbps and 2.5Mbps, very few people are interested in using them. 31.25kbps Physical Layer has its niche to replace 4-20mA transmissions in plant floor and work in a hazardous area. Other Physical Layers are giving their places to the state-of-art technologies.

6.1 Why Ethernet?

Ethernet is one of the most popular networks for office and business applications. Network components such as cables, hubs and switches are available with very low prices from the commercial off the shelf (COTS) environment.

Its media are evolving very quickly. 100Mbps Ethernet is replacing traditional 10Mbps Ethernet, and a faster Ethernet of 1Gbps is emerging. It is better to utilize those COTS components both in cost and availability.

6.2 TCP/IP Protocol Suite

Much more important thing for high-speed fieldbus is TCP/IP, which is the *de facto* standard in information technology (IT) world. When FOUNDATION fieldbus goes over TCP/IP, it makes it possible to open the door to the IT world and build a total network from plant floor to enterprise level.

IP (Internet Protocol) is used to transfer data to a desired station IP address. FOUNDATION fieldbus utilizes IP version 4 today and will switch to much powerful IP v6 in the near future.

TCP (Transfer Control Protocol) provides connection-oriented transport services that can be used for Client-Server communication. UDP (User Data Protocol) transfers an amount of data to a desired application and can be used for Source-Sink communication. Publisher-Subscriber communication can use either TCP or UDP.

6.3 Field Device Access Protocol

It is desirable to have the same applications for measurement and control regardless of their physical location in a hierarchical network. FMS and SMKP services must be maintained for this goal.

Field Device Access (FDA) protocol is designed for this purpose to allow all FMS and SMKP services go over TCP/IP.

Part-B Applications

This part explains Function Blocks and other applications running over FOUNDATION fieldbus. Communication technologies supporting those applications are explained in Part-A of this textbook.

This part gives you the knowledge on how measurement and control applications are implemented on FOUNDATION fieldbus. You will understand how carefully such applications are designed so that they can give your plant secure control and successful maintenance.

7. Virtual Field Devices

A fieldbus device may have user applications, which are independent from each other and do not interact. A fieldbus device consists of Virtual Field Devices (VFDs) for such individual applications. VFDs can be seen as different field devices from an application point of view. Communication services guarantee their independence. See Figure A-16 for relationship between VFD and communication.

7.1 VFDs in a Fieldbus Device

A FOUNDATION fieldbus device has at least two VFDs.

One is the **Management VFD** where network and system management applications reside. It is used to configure network parameters including VCRs as well as to manage devices on a fieldbus.

The other is a **Function Block VFD** where Function Blocks exist. It is possible for a field device to have two or more Function Block VFDs.

8. Function Block

This chapter focuses the most important concept of Function Block in FOUNDATION fieldbus, especially its models and parameters, through which you can configure, maintain and customize your applications.

8.1 What is a Function Block?

A Function Block is a functional model common in measurement and control. It is a generalized concept of the functionality you have in field instruments and control system such as analog input and output as well as PID control. The FOUNDATION specification, "Function Block Application Process - Part 1," gives fundamental concepts while Part 2 and later give various Function Blocks details.

Function Block parameters are visible and accessible through communication services and the Block behavior depends on the values of parameters. A Function Block may reside in virtually any device on the network and a set of Function Blocks connected to each other to form an application can reside in one device or be distributed among devices. Fieldbus Foundation's System Architecture document says:

> *"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 Function Block VFD contains three classes of blocks: Resource Block, Function Block and Transducer Block.

8.1.1 Resource Block

A Resource Block shows what is in the VFD. It gives you the manufacturer's name, device name, DD and so on. If the VFD allows you to create or download a Function Block, Resource Block shows how much resource (memory and CPU time) is available. Status of hardware is also visible. Resource Block controls the overall device hardware and Function Blocks within the VFD.

8.1.2 Function Blocks

A Function Block is a generalized model of measurement and control. For example, the AI block conditions raw data from transducer(s) and provides the measured value in a common format. Function Blocks are classified into three classes: (1) a Standard Block as specified by the Fieldbus Foundation, (2) an Enhanced Block with additional parameters and algorithm, and (3) an Open Block or a Vendor-specific Block designed by individual vendors.

A Function Block has input, output and contained parameters. Data generated in a block is exposed in an output parameter, which can be linked to the input parameter of other Function Blocks.

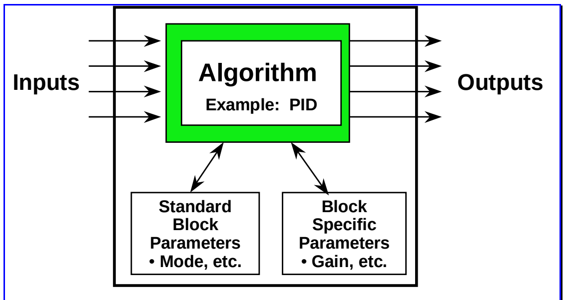


Figure B-1

Table B-1 lists Function Blocks defined by the Fieldbus Foundation. You can see major functionality is implemented here. Part 2 blocks are the most fundamental ones for measurement and control. Part 3 blocks are for advanced measurement. Advanced control blocks will be in Part 3 soon. Part 4 blocks provide I/O interface to other world such as 4-20mA. Part 5 blocks can be tailored for your application like a PLC program.

Part-2 Blocks	Major in control and measurement
AI	Analog Input Block
DI	Discrete Input Block
ML	Manual Loader Block
BG	Bias/Gain Station Block
CS	Control Selector Block
PD	P, PD Controller Block
PID	PID, PI, I Controller Block
RA	Ratio Station Block
AO	Analog Output Block
DO	Discrete Output Block
Part-3 Blocks	**Enhanced Blocks**
DC	Device Control Block
OS	Output Splitter Block
SC	Signal Characterizer Block
LL	Lead Lag Block
DT	Dead Time Block
IT	Integrator (Totalizer) Block
	(More blocks are under development)
Part-4 Blocks	**Multiple I/O Blocks**
MAI	Multiple Discrete Input Block
MDI	Multiple Analog Input Block
MAO	Multiple Discrete Output Block
MDO	Multiple Analog Output Block
Part-5 Blocks	**IEC61131 Blocks**
	(Under development)

Table B-1 – Function Blocks

Various measurement and control applications can be built by linking these Function Blocks. Figure B-2 shows typical examples using Part-2 blocks.

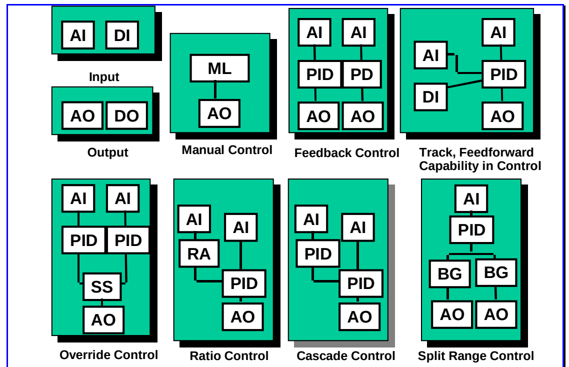


Figure B-2

8.1.3 Transducer Block

A Transducer Block is a model of sensors and actuators. It is modeled to give a similar expression to Function Blocks. Traditional sensors like pressure transmitters can be mapped to a Transducer Block. A Transducer Block is linked to a Function Block through the CHANNEL parameter of the Function Block.

A Function Block is a general idea while the Transducer Block is dependent on its hardware and principles of measurement. For example, pressure transmitter and magnetic flowmeter have different measurement principles but provide an analog measured value. The common part is modeled as an Analog Input (AI) block. The difference is modeled as Transducer Blocks that give you the information on the measurement principle.

8.2 Link and Schedule

A measurement or control application consists of Function Blocks connected to each other. Figure B-3 shows an example of PID control consisting of AI, PID and AO blocks. They are connected through “Link Objects” in Function Block VFD. A Link Object connects two Function Block within a device, or a Function Block to a VCR for Publisher or Subscriber.

A Function Block must get input parameters before its algorithm is executed. Its output parameters must be published after the algorithm execution. Therefore algorithm execution and Publisher-Subscriber communication must be orchestrated even when blocks are distributed among devices. The System Management and Data Link Layer cooperate to achieve this by using the Link Scheduling (LS) time distributed and synchronized by the Link Active Scheduler (LAS).

The System Management in a field device starts Function Blocks according to the Function Block schedule. The LAS transmits the Compel Data (CD) PDU to a publishing device to force the output data to be transmitted, according to the LAS schedule. These two schedules (Function Block Schedule and LAS schedule) are defined as offsets in the control period called "macro cycle," and must be configured to schedule Function Blocks and communication in a desired order. See Figure B-3 as an example.

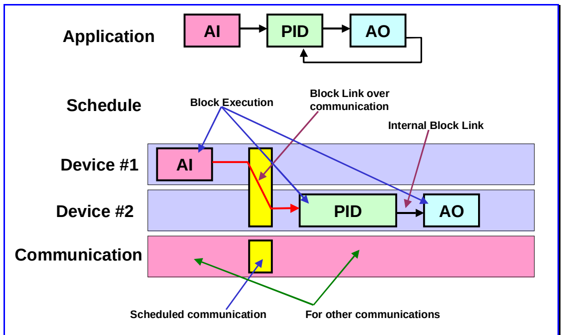


Figure B-3

8.3 Parameters

A block has a series of parameters, which are accessible by choosing one with a FMS index. Parameters of a block have continuous indices.

8.3.1 Parameter Classes

Block parameters are classified into three classes: input, output and contained parameters. Function Blocks can have all of them while the Resource Block and Transducer Blocks have only contained parameters.

8.3.1.1 Output Parameters

An output parameter is an output of a Function Block and can be connected to input parameter(s) of other Function Block(s). It is possible for two or more Function Blocks to share one output parameter from a Function Block. This is supported by periodic Publisher-Subscriber communications. An output parameter is a record consisting of a value (analog or discrete) and its status (showing whether the value is useful).

8.3.1.2 Input Parameters

An input parameter is an input of a Function Block and can accept one input parameter of another Function Block. Its data type must be equal to that of the output parameter.

8.3.1.3 Contained Parameters

A contained parameter is neither input nor output. It is accessible only through on-demand *Read* or *Write* request. Its data type can be any of those defined by the Fieldbus Foundation.

8.3.2 Parameter Attributes

Block parameters have several attributes that make their behavior complex.

8.3.2.1 Access Right

Function Block parameters can be readable (expressed as *r*) and/or writable (expressed as *w*). Even when a parameter is writable, there may be restrictions. For example, OUT parameter of AI block is writable when block mode is O/S or MAN. In other modes, a write request to this parameter is rejected. Vendor-specific range check for a write request is allowed. For example, many PID blocks reject a request to set proportional gain to zero.

8.3.2.2 Dynamic or Static

A dynamic parameter varies when the Function Block is executed. Its value is lost when the power is off. A static parameter does not vary by the block execution but may be written by an on-demand request. Its value is restored. If the value of a parameter just before power off is retained when the power is applied again, this parameter is called non-volatile.

Static parameters can be written only when block mode is O/S; e.g., you need to change block mode before modifying static parameters. Do not forget to bring the block back to appropriate mode (AUTO or CAS).

8.3.2.3 Status of a parameter

Input parameters, output parameters and some of contained parameters are records with status. Status shows whether the value of this parameter is useful or not. If the value is useful, the status is GOOD. If the value is not useful, the status is BAD. The status can be UNCERTAIN when the block is not 100% confident that the value is useful. Blocks have an option to interpret UNCERTAIN as GOOD or BAD. Status has additional fields to show more details.

8.4 Important Parameters

Several parameters are common to many blocks and are very important.

8.4.1 Block Mode

All blocks have their mode, expressed by the MODE_BLK parameter. It is a record of four components; *Target, Actual, Permitted* and *Normal*. *Target* is the mode into which an operator wants to bring this block. This component is writable. *Actual* shows the actual mode of the block and is read-only. When the necessary condition is satisfied, *Actual* mode becomes same as *Target*. There is a chance that *Actual* mode stays different from *Target* for some reason.

Permitted mode shows which mode is allowed in *Target* of this Function Block. *Normal* mode is a reminder for the operator to record the mode that an operator expects in normal conditions. These two components are writable but it is not a good idea to change them without reason.

Mode can be one of O/S, MAN, AUTO, CAS, RCAS and ROUT. In O/S (out of service) mode, the block does nothing but set parameter status to BAD. In MAN (Manual) mode, Function Block execution does not affect its output. In AUTO (Automatic) mode, the block works independently from upstream Function Blocks. In CAS (Cascade) mode, the Function Block receives the set point from an upstream Function Block. *Actual* mode of Function Blocks for output or control may become IMAN (initialize Manual) or LO (Local Override) according to the status of a downstream block or local operation.

Permitted mode depends on the block. Resource Block has only O/S and AUTO modes. Transducer Block may have O/S, MAN and AUTO modes.

8.4.2 Scaling Parameters

Some Function Blocks need scaling of the data as 0~100%. A scaling parameter is a record of four components: *EU@100%, EU@0%, Unit Code* and *Point Position*. *EU@100%* and *EU@0%* are values of 100% and 0%, respectively, in the engineering units. *Unit Code* indicates the engineering units of the scaling or parameter. It could be GPM, psi, inches or something else. *Point Position* shows the position of the point that an operator wants to show the desired resolution of the floating-point value. Table B-10 summarizes important unit codes for your convenience.

8.5 View Objects

Since a Function Block has many parameters, it is not practical to read them one by one for display purposes, especially for operators. Fieldbus Foundation utilizes the *FMS Variable List* concept for this purpose. A View Object is a predefined Variable List made of many block parameters.

Four View Objects are defined for each block. They are

- VIEW 1: Operation Dynamic – List of dynamic parameters that are necessary to run the plant by operators.
- VIEW 2: Operation Static – List of static (configured) parameters that may be necessary to show operators along with the dynamic parameters.
- VIEW 3: All Dynamic – List of all dynamic parameters that may be used for detailed display or diagnosis. This list is bigger than VIEW 1 object.
- VIEW 4: Other Static – List of static parameters that may be useful for configuration and maintenance purpose. This list is bigger than VIEW 2 object and may or may not consist of all static parameters.

9. Important Blocks

Fieldbus Foundation specified ten (10) standard function blocks in the Part 2 of its specification. More function blocks were added later. However, only five Function Blocks (AI, DI, PID, AO and DO) are of the most importance in most cases, and only three of them (AI, PID and AO) are in many cases.

This chapter gives you information of three important Function Blocks (AI, AO and PID) as well as Resource and Transducer Blocks. It also provides “recommended” values of block parameters, which come from Yokogawa’s long experience of measurement and control. You can use these values in most of cases. If a parameter is a bit-string (unsigned integer with each bit having an assigned meaning), names of bit to be set are listed using names in Foundation Document. If a parameter is dynamic and you cannot set it, value field indicates it as “dynamic.”

9.1 AI block

Analog Input (AI) block is designed to allow users to enjoy "standard" model of generalized signal-conditioning function. An AI block receives data measured by Transducer Block and manipulates it for

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

Figure B-4 shows the internal structure of an AI block.

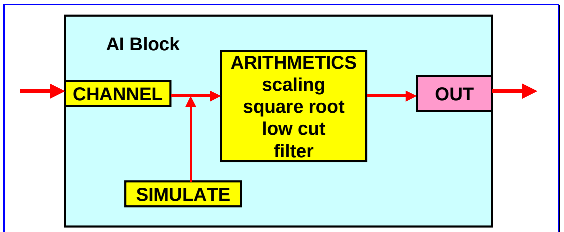


Figure B-4

9.1.1 Scaling and Square Root Calculation

Parameter L_TYPE controls scaling and square root. Value of a contained parameter PV (Process Value) of AI is determined by L_TYPE. When its value is "direct," Channel value becomes OUT value. When its value is "indirect," Channel value is scaled with XD_SCALE and OUT_SCALE.

XD_SCALE gives 0% and 100% value of Channel value and its engineering unit, while OUT_SCALE gives those of the output value.

When the value of L_TYPE is "Square Root," the output value is square root of scaled value. The resulting value could be very unstable because of the nature of orifice plate. In that case, a cutoff function is used to force PV value to zero when it is less than the LOW_CUT value.

9.1.2 Low-pass Filter

PV can be made more stable by applying single-exponential low-pass filter. Its time constant is given by PV_FTIME in seconds. If its value is zero, no filter is applied.

9.1.3 Alarm Generation

When PV value is smaller than LO_LIM or LO_LO_LIM, a LO or LO_LO alarm is generated, respectively. When PV value is larger than HI_LIM or HI_HI_LIM, a HI or HI_HI alarm is generated, respectively. Following order is expected:

LO_LO_LIM ≤ LO_LIM ≤ HI_LIM ≤ HI_HI_LIM

9.1.4 Permitted Modes

Permitted modes are O/S, MAN and AUTO. In MAN mode, you can modify OUT.*value*. In AUTO mode, PV.*value* and PV.*status* are copied to OUT.*value* and OUT.*status* respectively.

9.1.5 Channel Value

The CHANNEL parameter, 1 or greater (upper bound depends on Transducer Block), chooses one of the data values from Transducer Block. Channel value is visible in SIMULATE parameter (*Transducer Value* and *Transducer Status*).

9.1.6 Simulation

This functionality is very useful in starting up the plant. If you set SIMULATE.*EnDisAble* to Enabled, AI block uses *Simulate Value* and *Simulated Status* as Channel value instead of *Transducer Value* and *Transducer Status*. This function is active only when Simulation Switch (hardware) is ON. Do not forget to disable SIMULATE after using this function.

9.1.7 Recommended Parameter Values

Recommended values for the following the three applications in Table B-2 are listed in the Table B-7. Note that those values can be changed only when MODE_BLK.*Target* is Out-of-Service.

Application AI1: Simple Measurement
The Transducer Block gives measured value in the desired unit. AI block can be transparent. In case of Yokogawa devices, 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
Transducer Block gives 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
Transducer Block gives 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 B-2 – AI Block Applications

9.2 AO block

The Analog Output (AO) block is designed to allow users to enjoy a "standard" model of output devices like a valve positioner. An 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.

An AO block has bi-directional data-flow. One (forward path) is the flow from control value input to Transducer Block; the other (backward path) is the flow from Transducer Block to the control value. Figure B-5 shows the AO block structure.

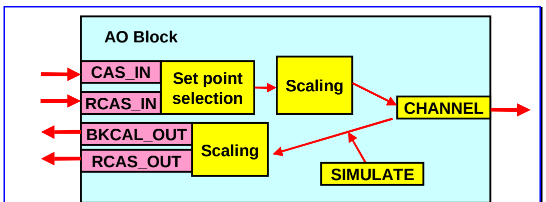


Figure B-5

9.2.1 Forward path

The control value from the controller becomes the set point, SP. AO block has several paths to calculate SP, depending on the block mode. In CAS mode, it is calculated from CAS_IN, subscribed from the publishing controller. In AUTO mode, SP value is given by on-demand *Write* request to SP. In RCAS (remote cascade) mode, remote controller gives 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 for the Transducer Block. In most cases those parameters have the same unit, e.g., %, and those scales can be simply 0 and 100 for *EU@0%* and *EU@100%.*

Before calculating SP, CAS_IN or RCAS_IN goes through limiting functions, boundary limit and rate limit. For boundary limit, the value is replaced with PV_HI_LIM (PV_LO_LIM) if the value is larger (smaller) than the limit. For rate limit, difference of current CAS_IN or RCAS_IN from the previous value is greater than SP_RATE_HI (or SP_RATE_LO to decrease); another value with limited rate is used to calculate PV.

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

OUT is given to the Transducer Block though CHANNEL.

9.2.2 Backward Path

Current position of actuator, like control valve, is manipulated through the backward path. Current position is fed to READBACK in transducer block unit (XD_SCALE) and then converted into the same unit of SP (PV_SCALE). Thus PV shows the valve position in SP unit.

BKCAL_OUT shows the current valve position (target or actual) to PID block. The value of SP (target position) is usually fed to BKCAL_OUT. It is possible to feed PV to BKCAL_OUT to include valve characteristics into the control loop (Note: this is not common in process automation).

9.2.3 Fault State

Fault State is prepared for very critical processes, where control valves need to be shut or open when the upstream block is unavailable (gives BAD status) for a pre-determined time interval.

9.2.4 Modes

An AO block can be in one of O/S, MAN, LO (Local Override), AUTO, CAS (cascade), RCAS (Remote cascade) and 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*.

9.2.5 Recommended Parameter Values

Recommended values for the following two applications in Table B-3 are listed in Table B-8. Note that those values can be changed only when MODE_BLK.*Target* is Out-of-Service.

Application AO1: Control with current target position
Use AO block with PID Function Block. Tracking value from AO block gives current SP given by PID or local operation.
Application AO2: Control with current valve position
Use AO block with PID Function Block. Tracking value from AO block gives current valve position, PV.

Table B-3 – AO Block Applications

9.3 PID block

A PID Block is a standardized model of PID control function. The process value comes through the IN parameter and its control output is the OUT parameter. Other input/output parameters exist to provide various control schemes, for example cascade control. Figure B-6 shows the PID block architecture.

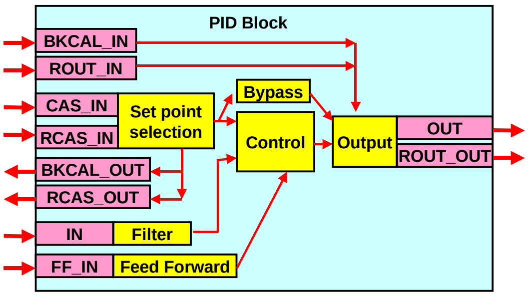


Figure B-6

9.3.1 Modes

A PID block can be in various modes, O/S, MAN, IMAN (Initialize Manual), LO (Local Override), AUTO, CAS (cascade), RCAS (Remote cascade) and ROUT (Remote output). Several new modes are important:

9.3.1.1 IMAN

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 to BKCAL_OUT, which is the current output of AO block.

9.3.1.2 *Normal* Mode

Normally the mode of the PID block is AUTO (closed loop) or CAS (cascaded loop). O/S and MAN may be used for operation.

9.3.2 Set Point

A PID block accepts the control set point via several methods depending on its mode. The following two parameters are important.

9.3.2.1 SP

You can directly write set point to SP parameter when MODE_BLK.*Target* is AUTO, MAN or O/S.

9.3.2.2 CAS_IN

In CAS mode, the PID block receives set point through the CAS_IN parameter. It must come from an upstream Function Block and the current set point is returned through BKCAL_OUT parameter.

9.3.3 PID Parameters

GAIN, RESET, and RATE are the tuning constants for the P, I and D terms, respectively. Gain is a dimensionless number. RESET and RATE are time constants expressed in seconds. It has another PID parameter, BAL_TIME, which you can be left zero (0).

9.3.4 Recommended Parameter Values

Recommended values for the following two applications in Table B-4 are listed in the table B-9. Note that those values can be changed only when MODE_BLK.*Target* is Out-of-Service.

Application PID1: Single Loop
PID block gives OUT to AO or another PID. An operator gives SP to this PID.
Application PID2: Cascaded Loop
PID block receives OUT from another PID

Table B-4 – PID Block Applications

9.4 Resource Block and Transducer Block

Detailed knowledge of Resource Block and Transducer Block is not always necessary. The only thing is their mode. Possible modes are O/S and AUTO. They affect the behavior of Function Blocks and must be set to AUTO for correct operation.

Transducer Blocks may have various parameters depending on their functionality. Their definition and behavior need to be discussed individually. However several parameters are common through Transducer Blocks.

Recommended values for Resource Block and Transducer Blocks are listed in Tables B-5 and B-6, respectively. Note that these values can be changed only when MODE_BLK.*Target* is Out-of-Service.

9.5 Recommended Values

Recommended values of block parameters are listed here. A number in front of a parameter name shows the index offset from the block header. Highlighted parameters (or components) are often “initialized on shipping” to useless values by some vendors. Those values prevent a Function Block from being operational and must be set to a meaningful value as listed in the tables.

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

Parameter Mnemonic	value
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

Notes:

*1) Do not set Fault State Supported bit of FEATURES_SEL unless you definitely need Fault State. See also note 1 for AO block in Table B-8.

Table B-5 – 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. 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

Parameter Mnemonic	value
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	

Notes:

*1) This table shows typical Transducer Block parameters. Since Transducer Block parameters depend on the physical principle, your Transducer Block may have different parameters. In any case, do not forget to set STRATEGY, ALERT_KEY and MODE_BLOCK to above values.

Table B-6 – Recommended Parameter Values of Transducer 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		

Table B-7 – Recommended Parameter Values of AI Block

Parameter Mnemonic	Application-1	Application-2
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-1	Application-2
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	Use 1 for experiment	
23. FSTATE_TIME	0 *1)	
24. FSTATE_VAL	0 *1)	
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	

Notes:

*1) This configuration is for typical applications where control valves should remain at the current position when the data from the controlling block becomes bad by communication error or block malfunction. If the process is very critical and the valve must come to a pre-defined position when the control data is not available, use Fault State option. Set Fault state to Value + Use Fault state Value in restart of IO_OPTS. Give the safety position to FSTATE_VAL and the time to go to Fault State in seconds to FSTATE_TIME. You must set the Fault State Supported bit in FEATURES_SEL of the Resource Block. See also note 1 for Resource block in Table B-5.

Table B-8 – Recommended Parameter Values of AO 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 *2)	
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	

Parameter Mnemonic	Application PID1	Application PID2
30. BKCAL_HYS	0.5	
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	

Notes:

*1) For Application PID2, add [SP-PV track in MAN+SP-PV track in LO or IMAN+SP-PV track in ROUT] to CONTROL_OPTS.

*2) Add [Target to next permitted mode if BAD CAS_IN] to STATUS_OPTS if your tool is able to show this new option (from FF 1.4).

Table B-9 – Recommended Parameter Values of PID Block

9.6 Unit Codes

Fieldbus Foundation has specified numeric codes for engineering units. Here is an extract of important ones.

It is much better to manipulate units with their "unit" text provided by DD instead of numeric expressions like this. They are listed here for your convenience in case your human-machine interface provides raw data expressions only. Refer to the Foundation document for missing units.

pressure

code	*unit*
1130	Pa
1032	MPa
1133	kPa
1034	mPa
1035	µPa
1036	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	second
1057	µs
1058	minute
1059	hour
1060	day

miscellaneous

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

Table B-10 – Major Unit Codes defined by Fieldbus Foundation (FF-903)

10. System Management

System Management is an important application for all Foundation devices. It manages device information and its behavior in a FOUNDATION fieldbus system.

10.1 Device Management

A device on FOUNDATION fieldbus can be identified from others by using one of three identifiers:

- Device Identifier (ID): A text string unique to the device in the world given by the manufacturer. It is burnt into the device and will never change.
- Physical Device (PD) Tag: A unique name in the plant or a set of fieldbus segments, assigned by the user. It is used to identify a device for a specific application purpose in the plant.
- (Physical) Node Address: A number of eight-bit length, unique in a fieldbus segment assigned by the user through network configuration.

Device ID is unique to the device and the same Device ID does not exist elsewhere in the world. It is given by the manufacturer and therefore good for management purposes.

PD Tag is assigned by the user to identify usage of the device in the plant. It is a text field 32 characters long. It is common practice to give the same PD Tag to a new device replacing old broken device.

Since Device ID and PD Tag are very long (32 bytes), it is not good to use them in daily communication, especially in the 31.25kbps low speed network. Instead, Node Address is used to identify devices in communication. Services to correlate these three identifiers are provided.

For example, a pressure transmitter is shipped with a permanent Device ID "59454300031999DEC22001102344" and configured to have PD Tag "FI1001" and node address 0xF5.

System Management Agent in a field device responds to System Management Kernel Protocol (SMKP) requests from a manager to configure the device. Its functionality is:

- To know information on a device at a specific address, including Device ID, manufacturer, device name and type,
- To clear and set the node address of the device using a specific Device ID,
- To clear and give a PD Tag to a device, and
- To find a device of specific PD Tag.

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

10.2 Function Block Management

As explained in section 8.2 of this textbook, the Function Block algorithm must start in a defined time. System Management Agent stores Function Block scheduling information and starts an assigned Function Block at the desired time. Macro Cycle is the period of overall application and the schedule is designed as an offset time from the start of macro cycle.

10.3 Application Time Management

All System Management Agents in a system keep an application time (or system time). It is used to record an event with a time stamp.

System time and Link Scheduling (LS) time are not equal. LS-time is local in the Data Link layer and is used for communication and Function Block execution. System time is more universal and is the same in all devices in a system consisting of multiple fieldbus segments.

11. Device Information Files

Applications such as human-machine interface and fieldbus configuration need more information of devices. Several files are standardized by the Fieldbus Foundation to make devices interoperable and help engineers.

11.1 Device Description

Device Description (DD) provides information on blocks. You can read a function block parameter by name and display it properly according to its data type and display specification. This is very useful in handling enumerated parameters. For example, you have only to choose "psi" instead of its value 1042. Whenever you get a new device, you are able to use its full functionality by simply installing its DD without updating host software.

You can run DD Method for a dedicated procedure (sequence of communications) for calibration, diagnosis and so on. DD Menu shows a list of DD Methods.

DD is useful for human-machine interface, system configuration and maintenance. You can control process without the help of DD.

- Gives information on parameters especially non-standard parameters
  - Name and explanation
  - Data structure and attribute
  - Access right
  - Help menu
  - Calibration methods, etc.
- Enables a standard screen show a new device

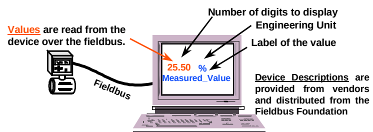


Figure B-7

11.1.1 What are DDL and DD?

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

The DD binary consists of two files. One is DD binary with the extension “.ffo” while the other is DD symbol list with the extension “.sym” . Once they are installed in your machine, you will have full access of the device.

It is not a good idea for device vendors to write full information individually. Instead, Fieldbus Foundation provides DD Library, which provides a common DD and dictionary. A device vendor has only to describe their special part with DDL. Because of this convention, DD binary files can be small.

Figure B-8 shows the hierarchical structure of DD. Fieldbus Foundation specifies the standard part of blocks and provides the DD “library.” Device “profiles” are under development to define a common part of various devices such as temperature transmitter and flow meter. The vendor-specific part is specified in their DD files.

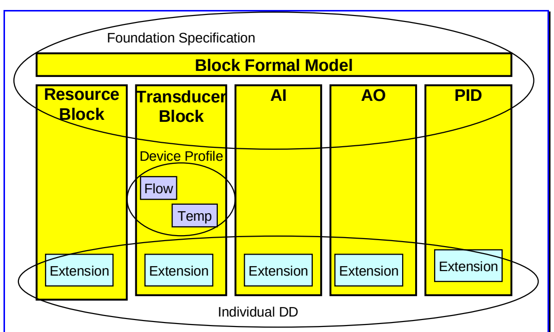


Figure B-8

11.1.2 Device Description Service

Device Description Service (DDS) is a software for your human-machine interface. It retrieves information in DD binary by using a key DD Item ID stored in the FMS Object Description of a parameter.

DD binary files are stored in the following directory structure:

<DD home directory>
+- Manufacturer ID
+--- Device type

Manufacturer ID is a unique code given to the manufacturer by the Fieldbus Foundation. You can identify the device manufacturer by this ID. It is thee byte in length and is shown in a hexadecimal expression with six characters. Yokogawa’s manufacturer ID is 594543, which stands for “YEC.”

Device type identifies the device from others of the same manufacturer by the two-byte data expressed in a four-character hexadecimal form. Yokogawa’s EJA has device type 0003.

11.2 Capabilities File

Capabilities File gives you information about the device capability of both Network/System Management and Function Block. Though certain parts of the information reside in the device itself, this file is useful for offline configuration where you can configure a fieldbus system without having real devices. A Capabilities file has an extension “.cff.”. Capabilities file is often called “CFF,” which stands for Common File Format.

Figure B-9 shows how common files are used during the system design and maintenance.

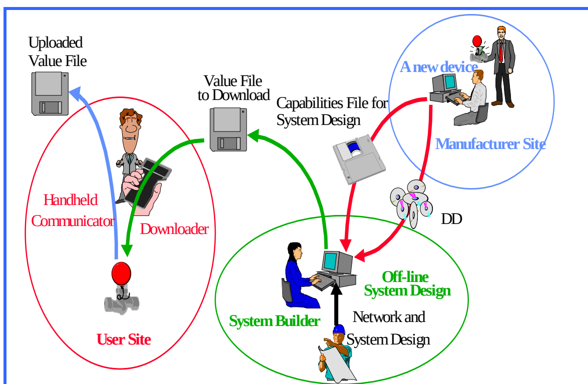


Figure B-9

Part-C Managing Fieldbus Projects

This part gives you an example of fieldbus project management. Again, fieldbus is an enabling technology and is not going to wipe out your expertise in projects. Major steps and procedures must remain as they were, and you have only to add knowledge and experience of fieldbus-based control and measurement.

Let us use an example project summarized in Figure C-1, which is a waste condensation plant of 15 feet × 20 feet, consisting of 12 transmitters and 2 control valves forming a single loop and a cascade loop. This part gives you key items for your consideration in managing this project. Though the selection of devices is one of the most important items to be considered, it is often beyond the technical consideration and therefore described very briefly in this textbook.

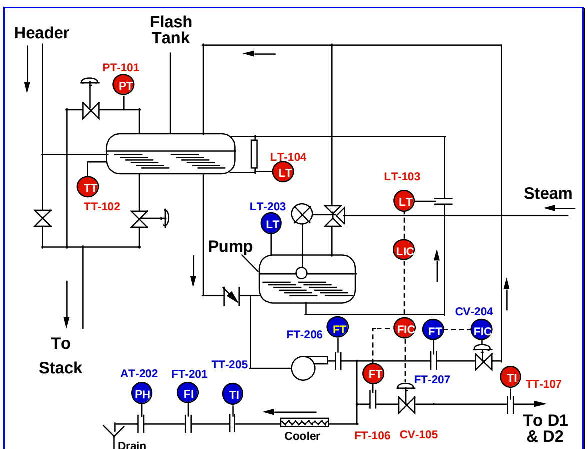


Figure C-1

12. Planning Phase

When you start planning of this project, you may want to minimize the initial cost of the project. Your plant will survive, however, for decades and you need to also consider the overall cost of the plant. Fieldbus is not an exception. You would be better to take a marginal design and reduce the future risk.

Let us count devices first. They are summarized as follows:

Number	Devices
4	Flow Transmitter
3	Level Transmitter
1	Pressure Transmitter
3	Temperature Transmitter
1	pH Transmitter
2	Control Valve

12.1 Devices on a Bus

How many devices do you put on a fieldbus segment? An economical analysis says twelve devices are enough to enjoy the initial cost reduction. It is also said four or five devices are profitable in a renovation project replacing the 4-20mA installation. Since our example project is a brand new plant, you may want to have all devices on a segment. Let us study the conditions to limit the number of devices before we reach the final decision.

Communication Specifications

The Physical Layer restricts the number of devices as described in the next section.

The Data Link Layer specifies the address to 8-bit length. Because some of these are reserved for special purposes, the actual address range for field devices is 232. This value is still large enough and you don't have to take it account.

Power Supply and IS Barrier

If a bus-powered device is hooked on the fieldbus, the necessary power must be fed through the fieldbus from a power supply somewhere on the network. The capacity of the power supply must be larger than the total amount of the average current consumption of all bus-powered devices on the fieldbus.

In the Intrinsic Safe fieldbus, the number of devices in the hazardous area is strictly limited by the total energy supplied through the safety barrier.

Since our example is not an I.S. installation and you have external power supplies of 350mA, this should not affect our decision.

Application Interaction

It is better to put interacting Function Blocks on one segment to avoid Function Block linkage over a bridge. A control loop is an example.

Application interaction needs communication services over the fieldbus. The more you have interactions, the more communication demands increase. If the communication throughput is not fast enough to transfer all the data, this would also limit the number of devices on the fieldbus.

A rough and safe estimation is as follows. Count the number of Function Block linkages in different devices (number of Publishers), N_P. Count the number of control blocks and output blocks (communication for Human-machine interface), N_C. Add two numbers to multiply 50ms.

$$T_{LOAD} = (N_P + N_C) \times 50ms$$

If the product in milliseconds is greater than 80% of the control period (macro cycle), this configuration is risky and you would better to remove devices. Detailed throughput analysis is available in the system design phase.

Let us study our project. We have 12 AI blocks, 2 AO blocks and 3 PID blocks. Assuming a control valve has a PID block and a level transmitter has another PID for cascade control, we get 13 (11+2) for N_P (11 AI blocks excluding one for cascade, and two-way communication between cascading PID blocks) and 5 for N_C (2 AO + 3 PID for control display). Since T_{LOA} is 900ms, we would be better to have two segments if we want controls in a second.

Risk Management
If a device on the fieldbus fails, there is a slight possibility that it destroys the segment communications. In the worst case, no measured values can be accessible through the fieldbus and the control activities may be interrupted. The same situation can happen by a loose wire or short circuits of the fieldbus cable. It is highly recommended to limit the number of control loops on a fieldbus segment to avoid a serious situation caused by this type of "worst mode failure."

Our Project
In the case of our project, we saw power supplies should not be the problem. Wiring should not be either because our plant is very small. Since we have a cascade control loop and a single loop, it is better to put them in different segments for risk management.

Interaction analysis suggests the same choice for two segments. It also says we could have 10 and 13 monitoring AI blocks within the throughput constraints, which should be enough for the future expansion.

Let us use two segments. The number of power supplies and host interface modules shall be 2. Now you can order all instruments including bus power supplies and terminators except for the cable.

12.2 Wiring Design

Your next step is to design wiring. Let us review the wiring design rules first.

Rule 1: Number of devices on a fieldbus is between 2 and 32.

Rule 2: Cable is a twisted pair of individual shield (type A) with 18 AWG wires.

Rule 3: Overall cable length does not exceed 1900m.

Rule 4: Overall spur length does not exceed 120m. Maximum number of devices decreases according to the spur length as shown in the table A-1.

Rule 5: When you use multiple twisted pairs with overall shield (type B), the total length decreases to 1200m.

We could use type B cable for trunk because all instruments are installed near the tank.

Measure the each spur length from the junction box to devices. Assign devices to the segment within the 120m limit of total spur length. Make sure devices to form control loops are on the same segment. We have two segments (red and blue in Figure C-1).

You are now able to order cables and their supplements (conduit, mating connectors, etc.). The trunk is type B cable while spurs are made of type A cable. Assign field devices tags to identify to which segment they should be connected. Figure C-3 shows this wiring.

12.3 System Design

Now you have a list of devices and available Function Blocks on each segment. Connect all AI Function Blocks except the ones on LI-103, FI-106, FI-207 to the monitoring application on the host

Control loops are combinations of AI, PID and AO blocks. Figure C-2 shows their connection for cascade level control. The PID block in the valve controls the flow (FIC110) while that in the level transmitter controls the level (LIC110). Target mode is also shown.

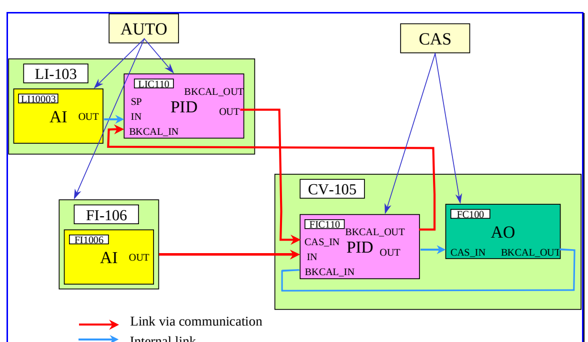


Figure C-2

Determine block parameters as explained in Part B of this textbook.

The configuration software generates other parameters for communication. Display the generated schedule to see how busy your fieldbus is.

13. Installation Phase

You can decrease the installation period by using the remote-configuration feature of fieldbus devices. Your engineers do not have to go to the field as frequently as they did with traditional 4-20mA devices.

13.1 Installation

Device installation is completely the same as for 4-20mA devices using installation and wiring drawings. It is recommended to assign PD tags to the devices before installation so that the device of the correct tag is installed at the correct location in the plant.

“Trunk” wiring is the cable with terminators in each end. The other part is called “spur” and should be within 120 m in total. Wiring from junction box to devices can be a tree (or “chicken-foot”) topology if the spur length is within this limit. Figure C-3 shows the wiring plan.

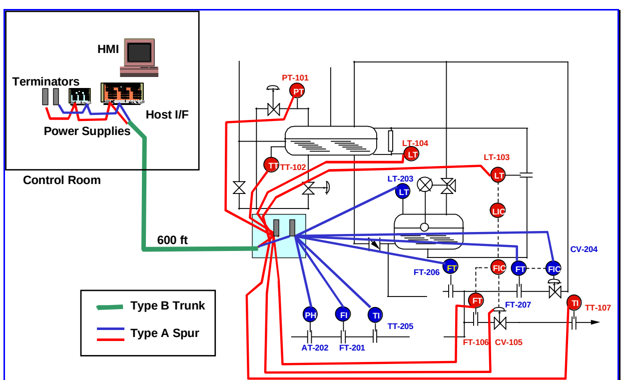


Figure C-3

13.2 Commissioning

After connecting devices to the wire, turn on the power supply to check whether all devices are operational. If they don’t have correct tag or address, assign them with the appropriate configuration software.

Download the configuration to all devices as well as the host interface module. All input values should be visible on the host screen.

Your plant may not be operational yet at the time of commissioning. Simulation function is useful for this purpose. Turn on the simulation switch in a device. The location of the switch should be explained in the device instruction manual. Write Enable to Simulate.*DisEnable* parameter of AI and AO blocks, then the blocks will use value and status that you write in the simulation field. You can now check that the display and control loops indicate the correct value as designed. Do not forget to disable the simulate hardware switch after the commissioning.

13.3 Startup and Test Operation

The instrumentation system over fieldbus is now operational. You can run the plant for a test. Input values may need damping for smooth recording. You are able to tune PID parameters of the loops.

After test operation and tuning, upload tuned block parameters so that you can use them if and when the device malfunctions.

14. Operation Phase

The operation phase should not change from traditional system.

14.1 Control Operation

During the operation phase, operators must be able to watch the plant operation and take over the control if needed.

Input monitoring, loop display, manual operations, and process alarm display are available through the host human-machine interface.

There is no change from traditional control and the operators need no additional training.

15. Maintenance Phase

Once the plant is in operation, it is in the maintenance phase at the same time. You should be able to maintain the fieldbus control system during plant operation. DD-based software is helpful for daily maintenance such as calibration and pH electrode cleaning.

15.1 Device Alarms

When a device detects a problem in its operation, it reports a “device alarm” to you. It might be the sensor burnt out, loss of air, a leak or something else.

Before you go to the device location, you can now see what the alarm is. Open your diagnosis window and see the reported alarms. The device may also provide more information in the Resource Block or Transducer Block. An electronic manual may be useful and the diagnostic software shows the details of the problem in text retrieved from the DD.

15.2 Replacing a Faulty Device

When you need a faulty device replaced, follow this procedure. Make sure the plant is safely operating even when a device malfunctioned.

Locate the replacing device in your stock. It must be the same device of the same revision. It is recommended to give it device tag and address before connecting it to the operating fieldbus. This clears all communication configurations and prevents any surprise in adding a device of unknown configuration to the operating fieldbus.

Disconnect the faulty device from the bus. Be careful not to short-circuit the bus. When the wire is connected with screw terminals, remove the screws as quickly as possible to avoid chattering as it causes momentarily undesired interrupt on the communications.

Replace the device and connect the wire again following the notes above. The device should start communications in seconds. When you set the tag and address before connecting the replacement, it joins the token rotation and stays inactive for control and measurement.

Download the configuration as well as the tuned parameters to the device to recover the operating condition. The applications using this device will be operational. Make sure the whole plant comes back to the normal state.

15.3 Asset Management

You may want to have information of devices in the field. When you do any maintenance, you keep records of what you did to the device. When a device malfunctioned, you need to find a replacement.

A management software package for such information on your asset is often very helpful. It is also necessary to plan the plant shutdown for preventive maintenance. Figure C-4 shows an example of such a management package.

16. Renovation Phase

When you plan to renovate your plant, you can shorten the plant shutdown period by using the online upgrade feature of fieldbus.

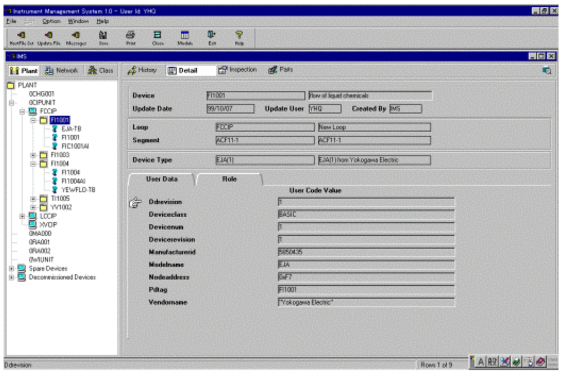


Figure C-4

16.1 Addition of Applications

Suppose you plan to add temperature measurement at two points to this plant. Your power supply has enough capacity for additional devices. Since you designed wiring with a good margin, you can add two transmitters to the existing fieldbus segments.

Install them on the plant and connect them to the fieldbus using additional spurs with the same care as for device replacement. Add them to your applications and download configurations to the devices. If the device addition does not change the control strategy, this addition does not affect other devices and therefore you can add them without disturbing the plant operation.

16.2 Device Upgrade

Suppose you want to upgrade existing “smart transmitter” to fieldbus and add that to the existing fieldbus segments.

The device vendor may provide an “upgrade kit” from smart to fieldbus. In that case, you don’t have to remove the installed instrument from the plant. Just replace the necessary electronics parts for the upgrade. Then you can add this device as an additional device to the fieldbus as explained in the previous section.

Several fieldbus devices support field software upgrade so you can have more functionality by this upgrade. Follow the instruction from the device vendor for this operation.

Acronyms

AC	Alternating Current
AI	Analog Input Function Block
APL	Application layer
AO	Analog Output Function Block
ASN.1	Abstract Syntax Notation 1
AUTO	Automatic mode
AWG	American Wire Gauge
BG	Bias/Gain Station Block
CCITT	International Telegraph and Telephone Consultative Committee
CAS	Cascade mode
CD	Compel Data DLPDU
CF	Capabilities File
CFF	Common File Format
CL	Claim LAS DLPDU
COTS	Commercial off the Shelf
CPU	Central Processing Unit
CS	Control Selector Function Block
CT	Compel Time DLPDU
DC	Direct Current
DC	Disconnect Connection DLPDU
DC	Device Control Block
DCS	Distributed Control System
DD	Device Description
DDL	Device Description Language
DDS	Device Description Service
Device ID	Device Identifier
DI	Discrete Input Function Block
DLCEP	Data Link Connection End Point
DLL	Data Link Layer
DLSAP	Data Link Service Access Point
DT	Data Transfer DLPDU
DT	Dead Time Function Block
DO	Discrete Output Function Block
EC	Establish Connection DLPDU
EU	Engineering Unit
EUC	End User Council
ETS	Enterprise Technology Solutions
FAS	Fieldbus Access Sublayer
FB	Fieldbus
FB	Function Block
FCS	Frame Check Sequence
FDA	Field Device Access
FF	Fieldbus Foundation
FMS	Fieldbus Message Specification
Gbps (Gbit/s)	Gigabit per second
HMI	Human Machine Interface
ID	Identifier
IEC	International Electro-technical Commission
IMAN	Initialize Manual mode

IP	Internet Protocol
IS	Intrinsic Safety
ISA	International Society of Measurement and Control
ISO	International Organization of Standard
IT	Information Technology
IT	Integrator Function Block
Kbps (kbit/s)	kilobits per second
kHz	kilohertz
LAN	Local Area Network
LAS	Link Active Scheduler
LL	Lead Lag Function Block
LM	Link Master
LO	Local Override mode
LS Time	Link Scheduling Time
mA	Milliampere
MAC	Medium Access Control
MAI	Multiple Analog Input Block
MAN	Manual mode
MAO	Multiple Analog Output Block
Mbps (Mbit/s)	Megabit per second
MDI	Multiple Discrete Input Block
MDO	Multiple Discrete Output Block
ML	Manual Loader Function Block
NM	Network Management
NMIB	Network Management Information Base
OD	Object Dictionary
OS	Output Splitter Block
O/S	Out of Service mode
OSI	Open System Interconnect
PC	Personal Computer
PCI	Protocol Control Information
PD	PD Control Function Block
PDU	Protocol Data Unit
PD Tag	Physical Device Tag
PHL (PHY)	Physical Layer
PID	PID Function Block
PLC	Programmable Logic Controller
PN	Probe Node DLPDU
PR	Probe Response DLPDU
PT	Pass Token DLPDU
PV	Process Value
RA	Ratio Station Block
RB	Resource Block
RCAS	Remote Cascade mode
RI	Request Interval DLPDU
ROUT	Remote Output mode
RQ	Round-trip Time Query DLPDU
RR	Round-trip Time Response DLPDU

RT	Return Token DLPDU
SC	Signal Characterizer Block
SDU	Service Data Unit
SM	System Management
SMIB	System Management Information Base
SMKP	System Management Kernel Protocol
SP	Set Point
SP50	Standard and Practice committee 50
TB	Transducer Block
TCO	Total Cost of Ownership
TCP	Transfer Control Protocol
TD	Time Distribution DLPDU
TL	Transfer LAS DLPDU
UDP	User Data Protocol
VCR	Virtual Communication Relationship
VFD	Virtual Field Device
VIEW	View Object
WAN	Wide Area Network

Revision Information

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

May 2001/1st Edition
Newly published

Written by	Fieldbus Gr. Marketing Dept. System Business Division
Published by	Yokogawa Electric Corporation 2-9-32 Nakacho, Musashino-shi, Tokyo 180-8750, Japan
Printed by	Yokogawa Graphic Arts Co., Ltd.