

Fieldbus —

Part 1: Guide to system architecture

ICS 25.040.40; 35.100

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI
Documents

AVA

Document Sharing Hub

Committees responsible for this Draft for Development

The preparation of this Draft for Development was entrusted to Technical Committee AMT/7, Monitoring and control aspects of advanced manufacturing technology, upon which the following bodies were represented:

- ABB Kent plc
- Cegelec Industrial Controls Ltd.
- Department of Trade and Industry
- Fisher-Rosemount Ltd.
- Foxboro Great Britain
- Institution of Electrical Engineers
- Measurement Technology Ltd.
- Process Automation and Computer Systems Ltd.
- Schneider Ltd.
- Siemens Automation plc

This Draft for Development, having been prepared under the direction of the Advanced Manufacturing Technology Assembly, was published under the authority of the Standards Board and comes into effect on 15 October 1996

© BSI 03-2000

The following BSI reference relates to the work on this Draft for Development: Committee reference AMT/7

ISBN 0 580 26494 7

Amendments issued since publication

Amd. No.	Date	Comments

Contents

	Page
Committees responsible	Inside front cover
Foreword	iv
1 Scope	1
2 Overview	1
2.1 Key Components of Fieldbus Systems	1
2.1.1 Function Block Application Processes	1
2.1.2 Object Dictionary (OD) and Device Descriptions (DD)	2
2.1.3 Network Communications	3
2.1.4 Network Management	3
2.1.5 System Management	4
2.2 Relationship to the ISO OSI Reference Model	5
2.3 Relationship to the International Electro-technical Commission (IEC) Fieldbus Standards	5
2.4 DD 238 Documents	5
3 Network Topology	6
3.1 Single Link Topology	6
3.2 Bridged Networks	7
3.3 Device Architecture	9
4 Application Process (AP)	9
4.1 Overview	9
4.2 Network Visible Objects	10
4.3 Application Layer Interface	10
4.4 Object Dictionary	10
4.5 AP Directory	11
4.5.1 AP Directory Header	11
4.5.2 Object References	12
4.5.3 AP Architecture Summary	13
5 Function Block APs	13
5.1 Overview	13
5.2 Block Types	15
5.2.1 Resource Blocks	16
5.2.2 Transducer Blocks	16
5.2.3 Function Blocks	16
5.2.4 View and Trend Objects	16
5.2.5 Alert Objects	16
5.2.6 Link Objects	17
5.3 Types of Function Blocks	17
6 System Management Kernel (SMK)	17
6.1 Overview	17
6.2 SMK Communications	18
6.3 SMK Services	18
6.3.1 SMIB Access	18
6.3.2 Tag and Address Assignment	19
6.3.3 Device Identification	20
6.3.4 Locating Remote Devices and Objects	20
6.3.5 Clock Synchronization	20
6.3.6 Function Block Scheduling	21
7 Communications Entity	21
7.1 Overview	21

	Page
7.2 Network Management Agent	22
7.3 FMS	22
7.4 Fieldbus Access Sublayer (FAS)	24
7.5 Data Link Layer (DLL)	25
7.5.1 DLL Lower Level (Media Access) Functions	25
7.5.2 DLL Upper Level (Data Transfer) Functions	27
7.6 Physical Layer	28
8 VCR and SMKP Operation	28
8.1.1 Report Distribution VCRs	29
8.1.2 Publisher/Subscriber VCRs	31
8.1.3 Client/Server VCRs	33
8.1.4 System Management Kernel Protocol (SMKP) Communications	35
9 Specification of Fieldbus Systems	36
9.1 Common Model	36
9.2 Device Descriptions	36
9.2.1 Device Description Language (DDL)	36
9.2.2 Device Description Services	38
9.3 Communications Profiles	38
9.3.1 Passive Devices	39
9.3.2 Remote Input/Output (I/O) Devices	39
9.3.3 Temporary Devices	39
9.3.4 Field Devices	39
9.3.5 Display and Control Devices	39
10 Configuration Information	40
10.1 Level 1 Configuration: Manufacturer Device Definition	40
10.2 Level 2 Configuration: Network Definition	40
10.3 Level 3 Configuration: Distributed Application Definition	41
10.4 Level 4 Configuration: Device Configuration	42
11 System Configuration and Startup	43
11.1 Offline Configuration	43
11.2 Link and System Startup	43
11.3 LAS Schedule Loading	44
11.4 Configuration Updates	44
Annex A Standard Data Types and Data Structures	45
Figure 1 — Key Components of DD 238 System Architecture	2
Figure 2 — Relationship of the DD 238 Communication Architecture to the ISO OSI Reference Model	5
Figure 3 — Simple, Single-Link Fieldbus Network	6
Figure 4 — Bridged Network Topology	7
Figure 5 — Spanning Tree Representation	8
Figure 6 — DD 238 Device Architecture	9
Figure 7 — Application Process Structure	10
Figure 8 — AP and Composite Object References	12
Figure 9 — Organization of Composite List References in the Directory	13
Figure 10 — Relationships between Application Process Components	14
Figure 11 — Function Block Linkages	14
Figure 12 — Function Block Internal Structure	15
Figure 13 — Block Composite Object	16

	Page
Figure 14 — Server View of the System Management Kernel Architecture	18
Figure 15 — DLL Node Addresses	19
Figure 16 — Fieldbus Communications Entity Architecture	21
Figure 17 — Report Distribution VCR Operation	30
Figure 18 — Publisher/Subscriber VCR Operation	32
Figure 19 — Client/Server VCR Operation	34
Figure 20 — SMKP Operation	35
Figure 21 — Generation and Use Of DDs	37
Figure 22 — Access to the Object Dictionary and its associated DD	37
Figure 23 — Field Device DDs are Composed of Standard and Manufacturer-Specific Component Definitions	38
Figure 24 — DD Services Use DDs to Provide Device Independent Access	39
Table 1 — Object Directory Header	11
Table 2 — FMS Objects and Services	23
Table 3 — FAS VCR Endpoint Characteristics	24
Table 4 — FMS Standard Data Types	45
Table 5 — Standard Data Structures	46

Foreword

This part of DD 238 has been prepared by Technical Committee, AMT/7. It supersedes DD 236-1:1996, which is withdrawn.

Portions of this document are owned by Fieldbus Foundation and are being used with their expressed written permission.

This publication contains the following parts:

- *Part 1: Guide to system architecture*
(Equivalent to FF-800 Revision 1.0M¹);
- *Part 2: Specification for 31.25 kbit/s Physical layer profile*
(Equivalent to FF-816 Revision 1.0¹);
- *Part 3: Specification for Data link layer services subset*
(Equivalent to FF-821 Revision 1.0¹);
- *Part 4: Specification for Data link protocol*
(Equivalent to FF-822 Revision 1.0¹);
- *Part 5: Specification for Fieldbus access sublayer (Services and protocol)*
(Equivalent to FF-875 Revision 1.0¹);
- *Part 6: Fieldbus Message Specification*
(Equivalent to FF-870 Revision 1.0¹);
- *Part 7: Fieldbus management;*
- *Section 7.1: Specification for System management*
(Equivalent to FF-880 Revision 1.0¹);
- *Section 7.2: Specification for Network management;*
(Equivalent to FF-801 Revision 1.0¹).

This Draft for Development is not to be regarded as a British Standard.

It is being issued in the Draft for Development series of publications and is of a provisional nature because the subject matter is still under development requiring wider exposure. It should be applied on this provisional basis, so that information and experience of its practical application may be obtained.

A review of this Draft for Development will be carried out not later than 2 years after its publication. Notification of the start of the review period, with a request for the submission of comments from users of this Draft for Development, will be made in an announcement in the appropriate issue of *BSI Update*.

According to the replies received, the responsible BSI Committee will judge whether the Draft for Development can be converted into a British Standard or what other action should be taken.

Observations which it is felt should receive attention before the official call for comments will be welcomed.

These should be sent to the Secretary of BSI Technical Committee AMT/7 at BSI, 389 Chiswick High Road, London, W4 4AL, giving the document reference and clause number and proposing, where possible, an appropriate revision of the text.

¹) Available from the Fieldbus Foundation, 9390 Research Boulevard, Suite II-250, Austin, Texas 78759, USA.

Disclaimer of warranties

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

Summary of pages

This document comprises a front cover, an inside front cover, pages i to vi, pages 1 to 46 and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

1 Scope

This document specifies the architecture of the DD 238 Fieldbus. It describes how the components specified in the other Parts of DD 238 may be related and configured to construct and support DD 238 Fieldbus devices, networks, and systems.

2 Overview

A fieldbus system is a distributed system composed of field devices and control/monitoring equipment integrated into the physical environment of a plant or factory. Fieldbus devices work together to provide I/O and control for automated processes and operations. Fieldbus systems may operate in manufacturing and process control environments that include intrinsic safety requirements. In these environments, devices operate with limited memory and processing power and with networks that have low bandwidth. The DD 238 System Architecture provides a framework for describing these systems as a collection of physical devices interconnected by a fieldbus network. Fieldbus networks may be composed of one or more links interconnected by bridges. Bridges transfer messages transparently between links. Communication between physical devices and across bridges is provided by physical layer and data link layer protocols.

Each physical device performs a portion of the total system operation by implementing one or more application processes. Application processes perform one or more time-critical functions, such as providing sensor readings or implementing control algorithms. These and other elementary field device functions are modeled as function blocks. Their activities are coordinated through configuration of their operating parameters, execution schedules, and communications.

NOTE Application Process (AP) is a term defined by the International Standards Organization (ISO) Open Systems Interconnect (OSI) Reference Model (RM), ISO 7498, to describe the portion of a distributed application that is resident in a single device. The term has been adapted to the fieldbus environment to describe entities within devices that perform a related set of functions, such as function block processing, network management, and system management.

Communication between application processes occurs through application layer protocols. The layers between the data link layer and the application layer, as defined by the ISO OSI RM, are not present in the DD 238 System Architecture.

2.1 Key Components of Fieldbus Systems

The DD 238 System Architecture is *simple* and *open*. It has been designed to be simple and to meet the functional, environmental, and technology needs of distributed manufacturing and process control systems. It is open so that distributed automation systems can be constructed from control and measurement devices manufactured by different vendors.

This architecture represents a departure from vendor specific control systems architectures that have dominated the marketplace in the past. In these architectures, proprietary interfaces define how devices communicate with each other. Devices that work in one vendor's control system, therefore, do not work in that of another. Furthermore, because of these proprietary interfaces, the upgrade and evolution of a system is controlled more by its vendor than by what the industry as a whole has to offer.

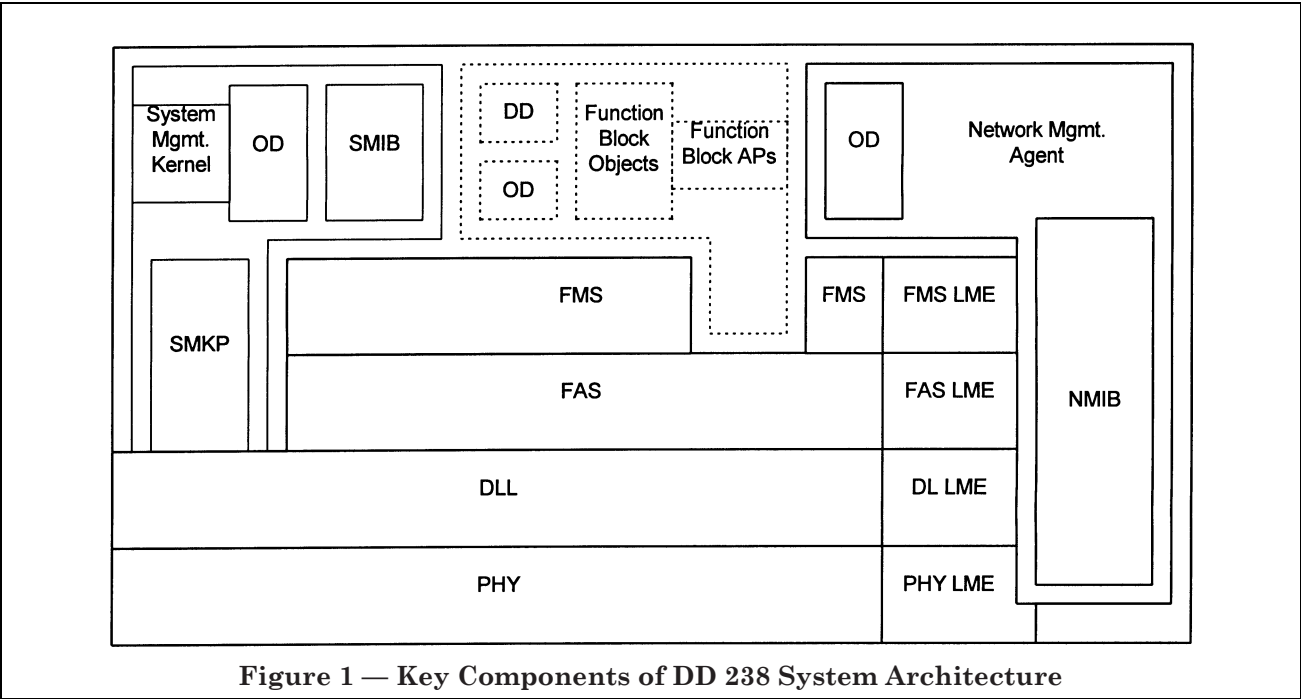
The DD 238 System Architecture has been designed to change this. Its purpose is to open the interface definitions to promote interoperability between and among devices, regardless of vendor. It addresses application and communications interfaces, and provides for network and system management. The basic structure of this architecture is illustrated in the figure below, followed by an overview of its key components.

NOTE DD 238 specifies the communication layers, shown respectively as *Phy*, *DLL*, *FAS*, *FMS*, and their Management components for System Management and Network Management. The other "User Layer" components are included and described as informative material to illustrate a full integrated system based on the DD 238 communication specification.

2.1.1 Function Block Application Processes

Automated manufacturing and process control systems perform a variety of functions. Because each system is different, the mix and configuration of functions are different. Therefore, the System Architecture has been designed to support a range of functional models, each addressing a different need.

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 Fieldbus Foundation has taken a first step in the standardization of function blocks by defining a small set of parameters used in all function blocks called *universal parameters*. A second level of standardization has occurred through the definition of a standard set of function block classes, such as input, output, control, and calculation blocks. Each of these classes also has a small set of parameters standardized for it. A third level of standardization provides definitions for transducer blocks commonly used with standard function blocks. Examples include temperature, pressure, level, and flow transducer blocks.

This classification scheme provides for a fourth level of standardization by allowing vendors to add their own parameters by importing and subclassing standard classes. This approach provides for the extensibility of function block definitions as new requirements are discovered and as technology advances.

2.1.2 Object Dictionary (OD) and Device Descriptions (DD)

Associated with the function block model are standardized tool definitions, that are used to support function blocks. Two of these tools are the Object Dictionary (OD) and the Device Description (DD). They provide for the definition and description of network visible objects of a device, such as function blocks and their parameters,

To promote consistency of definition and understanding of these objects, descriptive information, such as data type and length, is maintained in an Object Dictionary (OD) by device applications. The OD contains a description for each network visible object of an application, and makes this information available over the network.

OD descriptions of device applications can be supplemented by machine readable Device Descriptions (DD). DDs are written using the Device Description Language (DDL). The DDL is a programming language used to extend OD descriptions by providing *type* information about objects. The same DD type description may be used to describe a number of objects, located in either a single device or in multiple devices.

Once written, the DDL source is translated into a machine readable form, so that it may be loaded into the device it describes, or stored on an external medium, such as a floppy disk. Users of a device may then read its DD directly from the device, or obtain it from the external medium. The function block model is used in conjunction with the OD and DD to simplify the process of achieving device interoperability. Parameter transfers between function blocks can be easily verified, because all parameters are described using the OD and DD. Additionally, human interface devices do not have to be programmed specifically for each type of device on the network. Instead, their displays and their interactions with devices can be driven from OD and DD descriptions.

2.1.3 Network Communications

The DD 238 network transfers information between devices using preconfigured communications channels called Virtual Communications Relationships (VCR). Three types of VCRs are defined. Their characteristics are defined in Section 8.4 below, and their operation is described in Section 9 below.

- Publisher/Subscriber VCRs:** These VCRs are used by data producers (publishers) to post data into a *network buffer* that may be read, on demand, by users (subscribers) of the data. The network buffer is distributed among the publisher and its subscribers, such that a single copy of the data is maintained in all buffers. Updates to publisher buffers are transferred simultaneously to all subscriber buffers in a single broadcast. Transfers of this type can be scheduled on a precisely periodic basis. Using system management services, updates to the publisher buffer can be synchronized with the buffer transfers to reduce the delays between data production and data transfer.
- Report Distribution VCRs:** These VCRs are used to broadcast and multicast event and trend reports. The destination address of a VCR may be predefined so that all reports are sent to same address, or it may be provided separately with each report. Transfers of this type are queued. They are delivered to the receivers in the order transmitted, although there may be gaps due to corrupted transfers. These transfers are unscheduled and occur in between scheduled transfers at a given priority. A token passing mechanism is used to share the unscheduled time between devices.
- Client/Server VCRs:** These VCRs are used for request/response exchanges between pairs of applications. Like Report Distribution VCRs, their transfers are queued, unscheduled, and prioritized. However, unlike Report Distribution VCRs, they are flow controlled and employ a retransmission procedure to recover from corrupted transfers.

To support these VCRs, the DD 238 System Architecture defines a three layer communications architecture. First, the physical layer specifies how signals are sent. Second, the data link layer specifies how the network is shared and scheduled among devices. Third, the application layer defines the message formats available to applications for the exchange of commands and responses, data, and event messages. The components of the communications architecture are described in more detail in Section 8 below.

The Fieldbus physical layer is composed of the physical medium and signaling protocol used to transmit data. Two power options are defined for twisted wire pair. The bus powered option allows devices to draw their power directly from the bus. This option provides the ability to implement low power devices for use in intrinsically safe environments. The second power option is referred to as self-powered, and requires devices to draw their power from an external source. It does not provide power on the bus.

Three transmission speeds may be used in fieldbus networks, 31.25 kbit/s, 1 Mbit/s, and 2.5 Mbit/s. The 31.25 kbit/s and 1 Mbit/s speeds have been specified to support intrinsically safe environments.

2.1.4 Network Management

To provide a means for integrating the layer 2 and 7 protocols (the communication *stack* protocols) within a device, and for controlling and monitoring their operation, the DD 238 System Architecture includes a network management agent for each device. The network management agent provides capabilities that support configuration management, performance management, and fault management. The capabilities are accessed through the same communication protocols used to access other device applications, instead of requiring the use of a specialized network management protocol.

Using configuration management capabilities of the network management agent, parameters are set within the communication stack to support data exchanges with other devices in the system. This process normally involves defining the transfers between devices and then selecting the desired communications characteristics to support the transfers. These characteristics are loaded into the device using the configuration management capabilities of the network management agent.

As part of this configuration, the network management agent can be configured to collect performance and fault related information for selected transfers. This information is accessible during run-time, making it possible to view and analyze the behavior of device communications. If problems are detected, performance is to be optimized, or device communications are to be changed, then reconfiguration can be performed while the device is still operating. The nature of the reconfiguration depends upon whether or not communication with the other devices has been interrupted.

This configuration, performance, and fault information is contained in the Network Management Information Base (NMIB), although much of it actually resides within the communication stack itself. Like function block application information, the NMIB is represented by a *Management* VFD and described by an OD. The System Management Information Base also is represented by this VFD to provide a central point of access to management information within the device.

2.1.5 System Management

The FOUNDATION System Architecture includes a System Management Kernel (SMK) for each device. SMKs in all devices maintain information and a level of coordination that provides a distributed platform for the execution and interoperation of device applications. The information maintained by the SMK is defined as the System Management Information Base (SMIB) and is described by its OD.

One responsibility of the SMK is to provide for the configuration of basic system information into its SMIB, prior to device operation. A SMK startup procedure takes a device through a set of predefined phases as it prepares the device for operation.

During this procedure, a special system configuration device, such as a handheld, may pass system information to the SMIB through its standard fieldbus port. This permits the device to be configured in an off-line workbench network or on the operational network, depending on the particular needs of the system.

Once the device is configured, system management defines procedures for assigning it a permanent data link address, based on its configured *tag* (system specific name), and for bringing it to an operational state without affecting the operation of other devices on the network. A similar procedure permits temporary devices, such as hand-held terminals or portable workstations, to join and leave the network for short periods of time as needed.

After a device has been added to the network, its applications may find it necessary to locate remote devices and function blocks. To meet this need, the SMK provides a directory service to its applications. This capability may be used by an application to obtain network access information about an object by broadcasting its name to all devices on the network and waiting for the device containing the object to respond.

To become fully operational, the activities of devices and their function blocks may have to be synchronized with those of the other devices on the network. Two mechanisms are defined by system management to support this. First, system management provides for the synchronization of application clocks across devices to a common application time reference so that each device shares a common sense of time. Second, and related to the first, system management uses scheduling objects that control when function blocks execute. This ensures that each function block executes at the appropriate time in relationship to other function blocks in the system and in relationship to their scheduled data transfers.

SMKs communicate using two separate application layer protocols. FMS is used for access to the SMIB using a standardized Management VCR. This VCR is used also to access Network Management Information contained in the joint SM/NM VFD. A special purpose management protocol, the System Management Kernel Protocol (SMKP), is used to support all other functions of the SMK. This protocol is integrated into the SMK, and operates directly over the data link layer (no VCR is used).

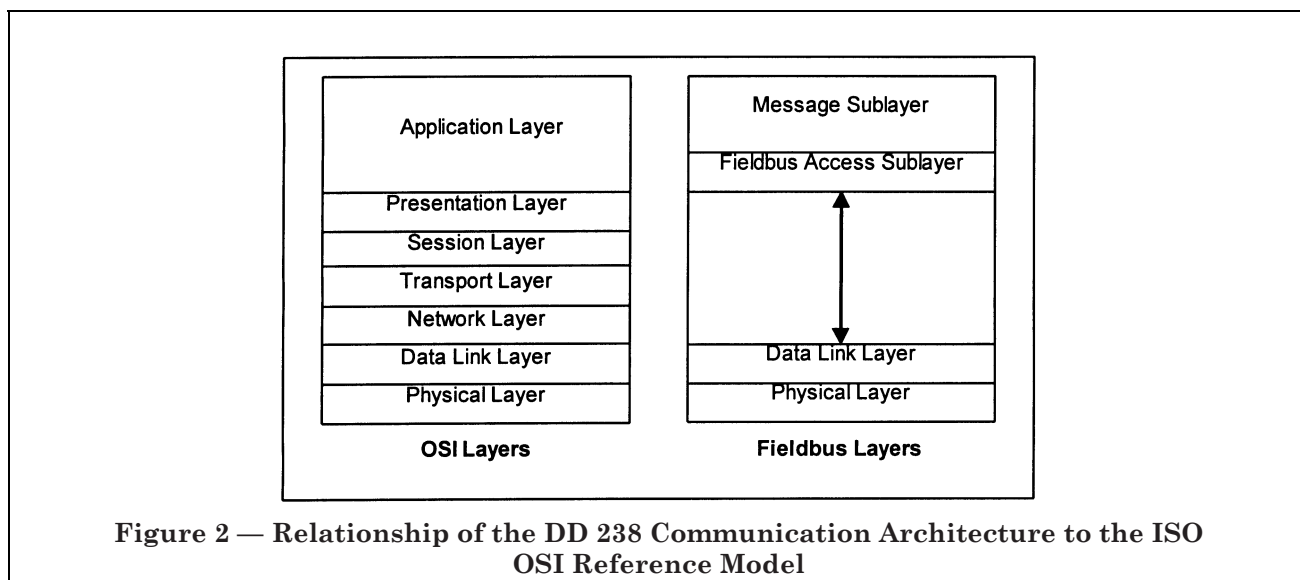
2.2 Relationship to the ISO OSI Reference Model

The ISO OSI Reference Model defines a general seven layer communications architecture for the interconnection of open systems. This general architecture has been optimized for the fieldbus environment by removing the middle layers (which are normally associated with general purpose, non-time critical applications such as file transfer and electronic mail). This three-layer architecture, and its relationship to the ISO OSI seven layer architecture is shown in the following figure.

As shown in the figure above, the Application Layer for the fieldbus is composed of two sublayers. The Message Sublayer provides communication services to fieldbus APs, system management APs, and to network management APs.

The Fieldbus Access Sublayer (FAS) provides different types of communication channels, called Virtual Communication Relationships (VCRs). VCRs are mapped onto the underlying network by the FAS to insulate application processes from changes in the network technology.

The middle layers of the OSI architecture are not present in the system architecture, while the Fieldbus Data Link and Physical Layer definitions conform to the OSI definitions for these layers.



2.3 Relationship to the International Electro-technical Commission (IEC) Fieldbus Standards

The DD 238 System Architecture is the same as that proposed in IEC SC65C WG6. The DD 238 Application Layer Protocols are similar to those being developed in the IEC SC65C WG6 Application Layer Committee. The Data Link Layer is a subset of the IEC 1158 Data Link Layer specifications. The DD 238 Physical Layer is a subset of the IEC 1158 Physical Layer specification.

2.4 DD 238 Documents

The other parts of the DD 238 define standards for user applications, network protocols, and network management. A summary of each of the documents in the DD 238 Specifications is provided below, with additional informative material about Fieldbus Foundation User Layer components (available from the Fieldbus Foundation, 9390 Research Boulevard, Suite II-250, Austin, Texas 78759, USA) suitable for making an integrated application based on DD 238.

Guide to System Architecture — provides an overview of the DD 238 Fieldbus. It describes the Architectural Model for Fieldbus Systems, Fieldbus Devices, and Fieldbus Communications, including the rules for naming and addressing.

Function Block Application Process (FBAP) — describes function blocks and the application processes that contain them.

Device Description Language (DDL) — describes the language used to specify fieldbus applications and their interactions. This is one of the tools used within the Fieldbus Foundation to achieve interoperability.

System Management (SM) — specifies the data, procedures, and protocols used to manage basic fieldbus device capabilities.

Network Management (NM) — specifies data and procedures used to manage the communications stack of a fieldbus device.

Fieldbus Message Specification (FMS) — specifies the DD 238 application layer services and protocols used by fieldbus applications. This protocol supports both client-server and publisher-subscriber (distributed data base services) models for direct control of applications and for distribution of data and events.

Fieldbus Access Sublayer (FAS) — specifies the services, protocols, and application layer communication relationship types used to convey application layer protocol data units. It also specifies the mapping of communication relationships to the data link layer.

Fieldbus Data Link Layer (DLL) Services Subset — specifies a subset of the IEC/CDV 1158-3 (65C/160A/CDV) Part 3 Data Link Service Specification.

Fieldbus Data Link Layer Protocol Subset — specifies a subset of the IEC/CDV 1158-4 (65C/161/CDV) Data Link Protocol specification.

Fieldbus Physical Layer Specification — specifies a subset of the IEC 1158-2:1993 Physical layer specification and service definition.

Fieldbus Communications Profiles — Separate Communications Profiles are identified for different operating environments. Each profile selects the communications capabilities required for a given environment and values for selected communications parameters for the environment.

3 Network Topology

This section describes the bus topologies used to interconnect fieldbus devices, and the internal architecture of fieldbus devices as seen from a network perspective.

3.1 Single Link Topology

The figure below illustrates a simple link fieldbus network consisting of a single link. Links with only two devices are typical of off-line configuration networks containing a configuration device, such as a handheld, and the device to be configured.

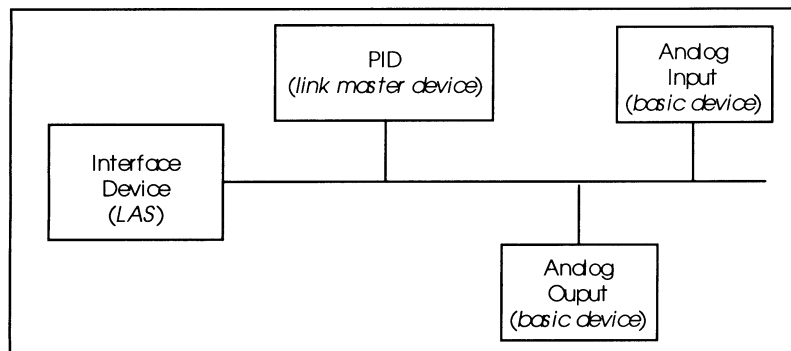


Figure 3 — Simple, Single-Link Fieldbus Network

All links are required to have one and only one Link Active Scheduler (LAS). The LAS operates at the data link layer as the bus arbiter for the link. The LAS:

- recognizes and adds new devices to the link.
- removes non-responsive devices from the link.
- distributes Data Link and Link Scheduling time on the link. Data Link Time is a network-wide time synchronized by the data link layer. Link Scheduling time is a link specific time represented as an offset from Data Link Time. It is used to indicate when the LAS on each link begins and repeats its schedule. It is used by system management to synchronize function block execution with the data transfers scheduled by the LAS.
- polls devices for buffered data at scheduled transmission times.
- distributes a priority-driven token to devices between scheduled transmissions.

Any device on the link may become the LAS, as long as it is capable. The devices that are capable of becoming the LAS are called *link master* devices. All other devices are referred to as *basic* devices.

When the link first starts up, or upon failure of the existing LAS, the link master devices on the link bid to become the LAS. The link master that wins the bid begins operating as the LAS immediately upon completion of the bidding process. Link masters that do not become the LAS act as basic devices when viewed by the LAS. They also act as LAS backups by monitoring the link for failure of the LAS, and by bidding to become the LAS when a LAS failure is detected.

In certain cases, it may be desirable to have the LAS role assigned to a specific link master. In these cases, one link master may be designated as the *primary* link master. If the primary link master fails to win the bid, it will direct the winning link master to transfer the LAS role to it. Once the LAS for a link becomes established, the link becomes operational.

3.2 Bridged Networks

Links of different speeds and media types may be bridged together to form multi-link, or *bridged*, networks. Data link layer bridges are used in these networks to interconnect links. In all DD 238 bridged networks, there is only one data path between any two devices. To ensure this, routing tables within bridges are coordinated with each other to form a *spanning tree*.

The spanning tree represents a configuration of bridges, such that there are only two directions of data flow, either towards the root of the tree, or away from it. There are no loops and no parallel routes. That is, for each link there must be one and only one active bridge leading to the root.

Each bridge in the spanning tree has a single root port and one or more downstream ports. Each bridge port represents a connection to one link. Root ports lead toward the root and the downstream ports lead away from the root. Downstream ports are also referred to as *designated* ports in the Data Link Layer Annex containing the bridge specification.

When a message destined for a remote link is received on the root port, the bridge always selects and routes the message to the appropriate downstream port(s) based on the routing tables within the bridge. When a message is received on a downstream port, the bridge routes it towards the root and/or away from the root on one or more of its other downstream ports. The figure below illustrates a bridged network and a spanning tree configuration for it.

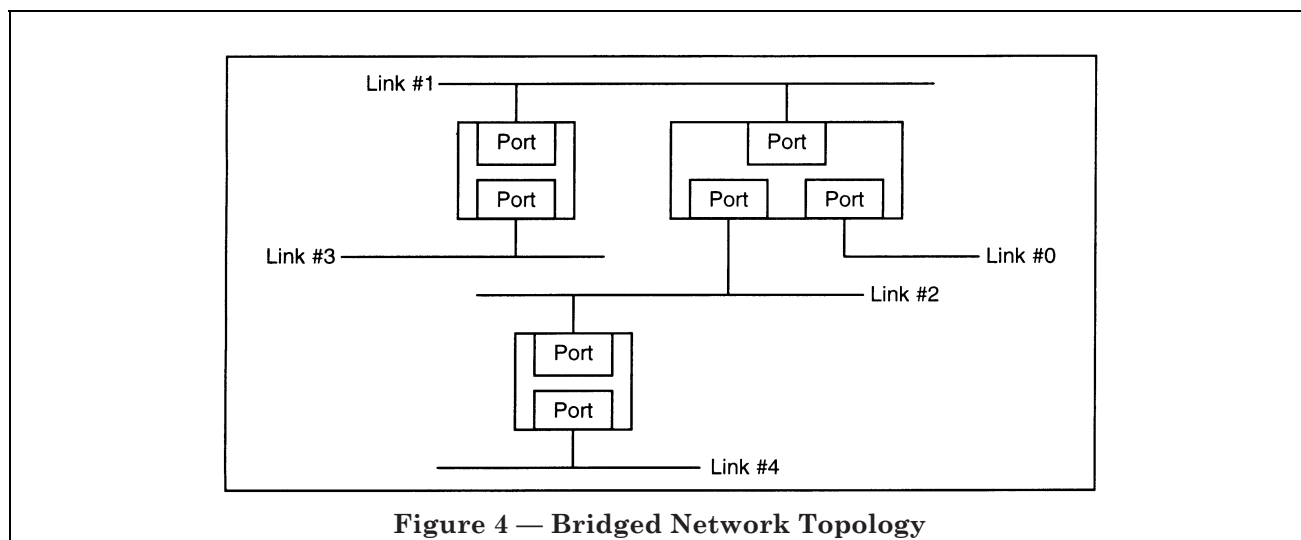


Figure 4 — Bridged Network Topology

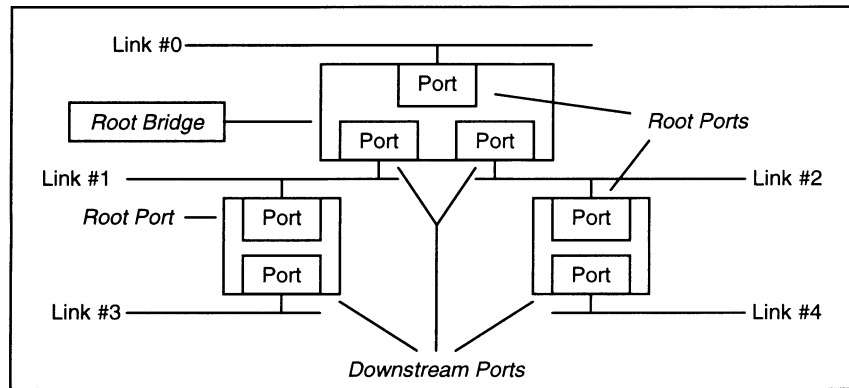


Figure 5 — Spanning Tree Representation

Bridge devices are responsible for performing four activities in DD 238 networks:

- 1) forwarding
- 2) republishing
- 3) data link time redistribution
- 4) application clock time redistribution

The first three are data link layer functions, and the fourth is a system management function. To support the last two, it is necessary for bridge ports to be the LAS of their downstream links. In the figure above, the root bridge contains the LAS for Links #1 and #2.

Forwarding occurs when a bridge receives a message with an address that the bridge has in its forwarding tables. If the address indicates that the message is to be forwarded, the bridge *appends* the message to the first-in first-out output queue for the appropriate port(s). When the bridge receives the right to transmit, called a *token*, from the LAS associated with that port, the bridge transmits entries from the queue in priority order.

Republishing occurs when the bridge and the LAS for each of the bridge's outbound links are configured to republish the data. The LAS on the inbound link triggers the publisher to transmit the data. When the bridge receives the data with the source address of the publisher, it buffers the message for scheduled transmission on the appropriate outbound ports as specified in its republishing table. The message is *republished* separately onto each of the outbound links when the associated ports are instructed to do so by their corresponding LASs.

It is possible for published data to be forwarded instead of republished. In this case, the address of the published data is configured into the forwarding tables, and not the republishing tables. This capability is provided to support unscheduled publishing. Unscheduled publishing occurs when a subscriber polls the publisher for the data instead of the LAS.

Redistribution of data link time messages occurs when a data link time message is received by a bridge on its root port. Data link time messages are sent from the data link time master located on the root link. When a bridge receives a data link time message, it recalculates the offset for the link scheduling time of each of its outbound ports for which it is the LAS. Once recalculated, the bridge queues the message for unscheduled transmission on each of its downstream ports. If the bridge contains downstream ports for which it is not the LAS, then the data link time cannot be redistributed to that link. In that case, that link, and all links downstream from it, operate using a data link time that is not synchronized with that distributed by the data link time master.

Redistribution of application clock messages occurs when an application clock message is received by a bridge on its root port. Application clock messages are sent from a system management time master. They contain the system time and the corresponding link scheduling time. When a bridge receives such a message, the system management kernel for each output port recalculates the link scheduling time for each of the destination links. Once recalculated, the message is queued for unscheduled transmission on each of its output ports.

3.3 Device Architecture

The fieldbus device architecture describes the internal structure of devices that are physically connected to the network medium. These devices are referred to as physical devices.

Each physical device, as illustrated below, is composed of a system management kernel, one or more application processes, and a communications entity. Interfaces between these components within a device are local and are not specified. Descriptions of each of these components are provided in the sections that follow.

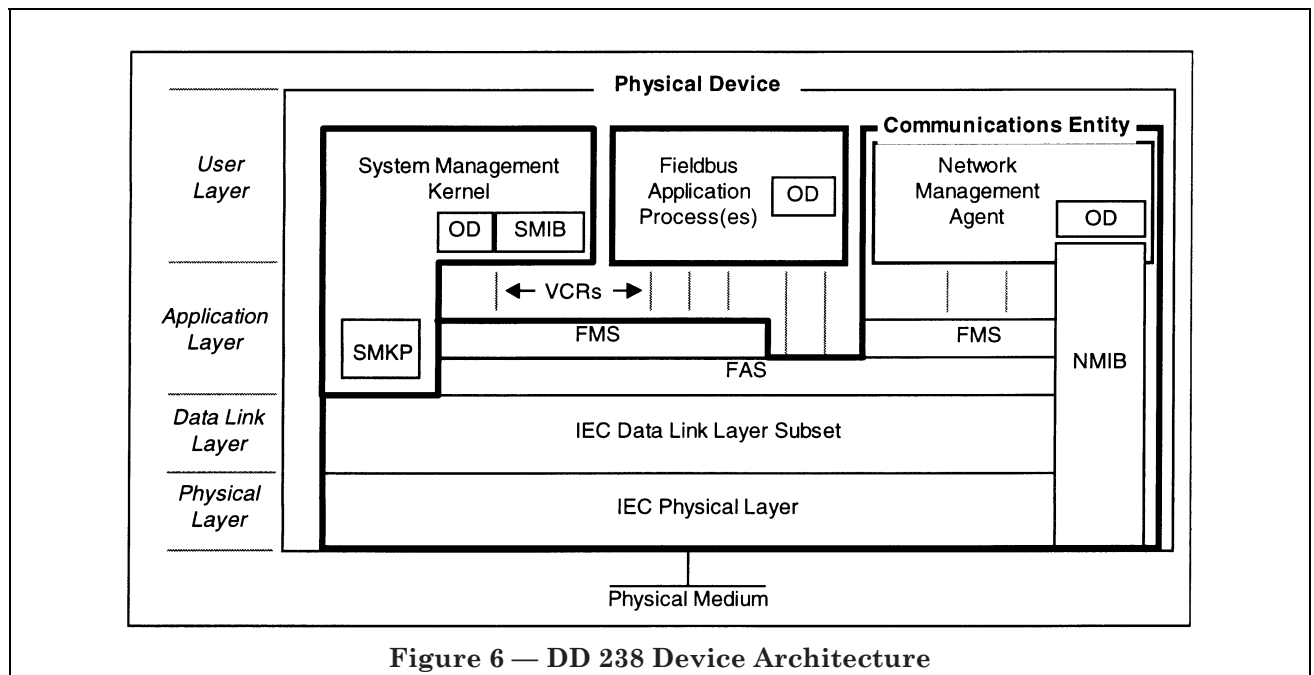


Figure 6 — DD 238 Device Architecture

4 Application Process (AP)

The Fieldbus AP Architecture supports distribution of time-critical and non-time-critical functions in a fieldbus system. Each Application Process (AP) represents a separately packaged set of functions that are allocated to a single device. The number of APs that a device may contain is implementation dependent. The DD 238 specifications place no restrictions on the number contained.

Whether or not an AP can be loaded into a device during device configuration or network operation is dependent on the physical capabilities of the device and how the AP has been implemented. Some physical devices, such as a Personal Computer (PC) or Programmable Logic Controller (PLC), can be implemented to receive their APs as software downloads. Other devices, such as simple transmitters or actuators, may have their APs implemented in application specific integrated circuits (ASIC).

This section specifies the internal structure of fieldbus APs and how they access the network. How they access each other within a device is local and is not specified.

4.1 Overview

Application processes are defined in ISO OSI Reference Model (ISO 7498) as the information and processing components of a distributed system (or application) that may be addressed and accessed over the network. From this viewpoint, fieldbus systems can be seen as a collection of cooperating APs.

The network visible portion of a fieldbus AP is composed of an AP Directory, an AP Object Dictionary, a set of Network Visible Objects, and an interface to the application layer communication services. This concept of the fieldbus AP is shown and described below.