FIELDBUS TUTORIAL

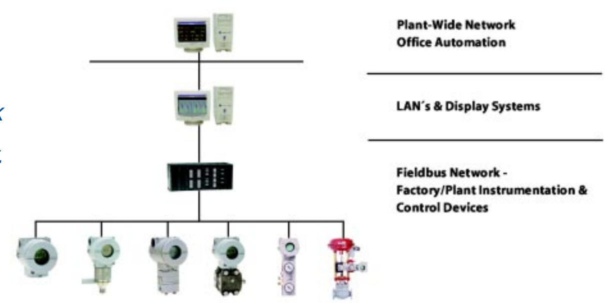
A FOUNDATION™ Fieldbus Technology Overview

Fieldbus - the future today.

FOUNDATION™ Fieldbus is the technological evolution to digital communication in instrumentation and process control. It differs from any other communication protocol, because it is designed to resolve process control applications instead of just transfer data in a digital mode. The Fieldbus technology is explained along this overview so you can check its advantages, feel its power and go a little further using fieldbus as an outstanding and reliable technology into your control system.

FOUNDATION™ Fieldbus is an all-digital, serial, two-way communication system, which interconnects "field" equipment such as sensors, actuators and controllers. Fieldbus is a Local Area Network (LAN) for instruments used in both process and manufacturing automation with built-in capability to distribute the control application across the network.

Plant Network Hierarchy.



Smar has already hundreds of fieldbus control systems installed worldwide. They make use of a full set of products and value-added software. Their highlights are the variety of field devices and the LC700 programmable logical controller with FOUNDATION™ Fieldbus module, enabling the association of the discrete and the analog worlds.
The control strategy is distributed along the field devices. It is possible because, besides having function blocks in their microprocessors, they have also ability to communicating fast and reliably to each other through the bus. From there comes also the fantastic flexibility of this technology. Devices can be networked and configured according to the user needs, being suitable from small systems to whole plants.
FOUNDATION™ Fieldbus is changing the concept of process management as an enabling technology. Thanks to all its additional power and great variety of new information, new tasks are made possible to automation professionals, such as new configurations, online performance diagnostic and maintenance records and tools.
Fieldbus is the Smar solution for the process automation of today.

TUTORIAL

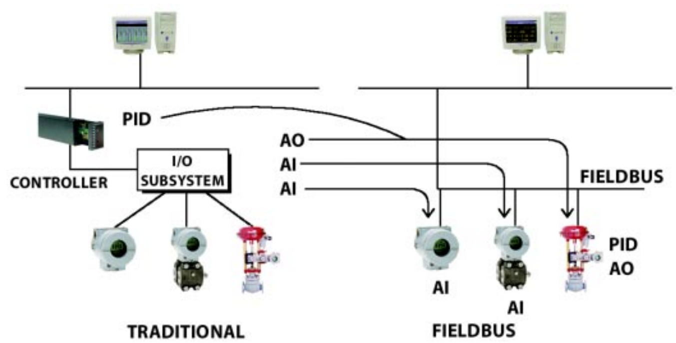
Fieldbus Benefits

Significant benefits are achieved in the control system life-cycle through the application of fieldbus technology.

Hardware Reduction

The FOUNDATION™ Fieldbus uses standard "Function Blocks" to implement the control strategy. Function Blocks are standardized automation functions. Many control system functions such as analog input (AlI, analog output (AO) and Proportional/Integral/Derivative (PID) control may be performed by the field device through the use of Function Blocks.

Hardware reduction.



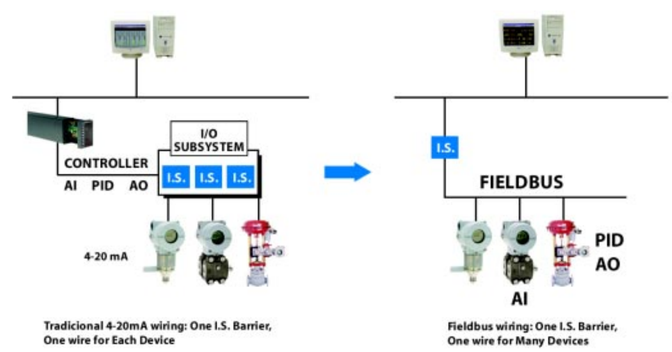
The consistent, block-oriented design of function blocks allows distribution of functions in field devices from different manufacturers in an integrated and seamless manner.

Distribution of control into the field devices can reduce the amount of I/O and control equipment needed including card files, cabinets, and power supplies.

Installation

The fieldbus allows many devices to be connected to a single wire pair. This results in less wire, fewer intrinsic safety barriers, and fewer marshaling cabinets

Installation Savings.



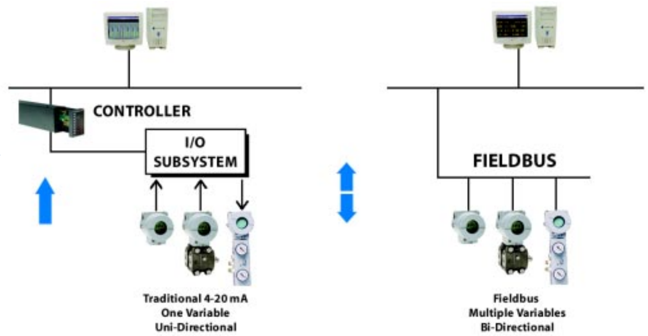
Data Quantity and Quality

In traditional automation systems, the amount of information available to the user did not go farther than the control variables. In FOUNDATION™ Fieldbus, the amount is much larger, due mainly to the facilities of the digital communication.

Besides that, fieldbus has increased resolution and no distortion (no A/D or D/A conversions), which gives more reliability to the control. All this added to the fact that the control is held within the field devices results in better loop performance and less degradation.

The fieldbus allows multiple variables from each device to be brought into the control system for archival, trend analysis, process optimization studies, and report generation. The high resolution and distortion-free characteristics of digital communications enables improved control capability, which can increase product yields.

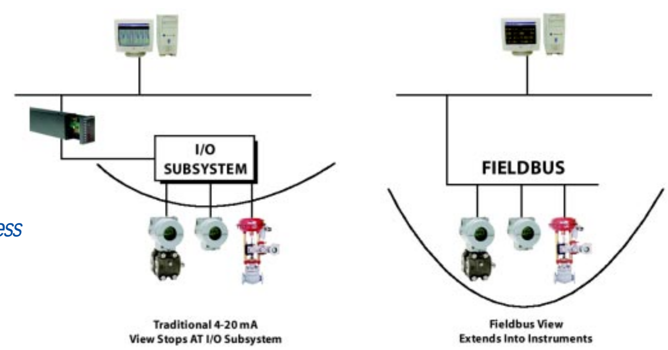
Fieldbus – Multiple variables, Both Directions



Maintenance

The self-test and communication capabilities of microprocessor-based fieldbus devices help reducing downtime and improving plant safety.
Upon detection of abnormal conditions or the need for preventive maintenance, plant operations and maintenance personnel can be notified. This allows corrective action to be initiated quickly and safely.

Expanded View of the process



Interoperability

FOUNDATION™ Fieldbus is also an open protocol, meaning that the FOUNDATION™ Fieldbus certified manufacturers are able to supply devices that will work together with devices from others certified manufacturers.
This "ability to operate multiple devices, independent of manufacturer, in the same system, without loss of minimum functionality" is called interoperability.

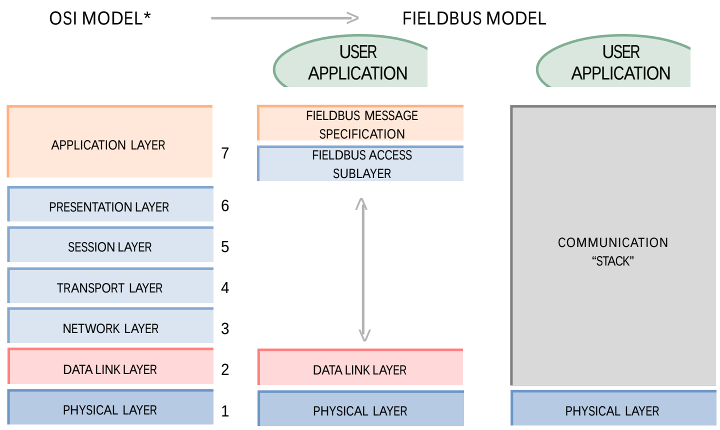
This flexibility to choose the supplier, knowing that all devices will work together is in fact a fantastic victory of all users.

FOUNDATION™ Fieldbus Technology

FOUNDATION™ Fieldbus technology consists of three parts:

1 – The Physical Layer;
2 – The Communication "Stack";
3 – The User Application.

The Open Systems Interconnect (OSI) layered communication model is used to model these components.



The Open System Interconnect (OSI) layered communications model.

* The user aplication is not defined by de the OSI Model.

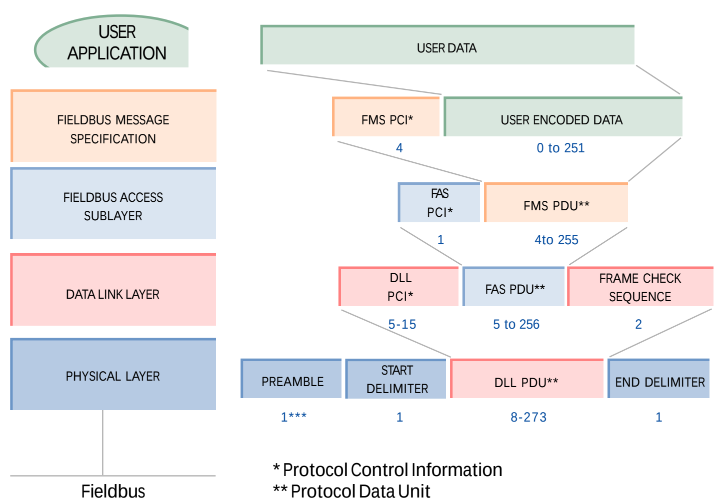
The Physical Layer is OSI layer 1. The Data Link
Layer (DLL) is OSI layer 2. The Fieldbus Message Specification (FMS) is OSI layer 7. The Communication Stack is comprised of layers 2 and 7 in the OSI model.

The fieldbus does not use the OSI layers 3, 4, 5 and 6. The Fieldbus Access Sublayer (FAS) maps the FMS onto the DLL.

The User Application is not defined by the OSI model. The Fieldbus Foundation has specified a User Application model.

Each layer in the communication system is responsible for a portion of the message that is transmitted on the fieldbus.

The numbers below show the approximate number of eight bit "octets" used for each layer to transfer the USER data.



The approximate number of eight bit "octets" used for each layer to transfer the USER data.

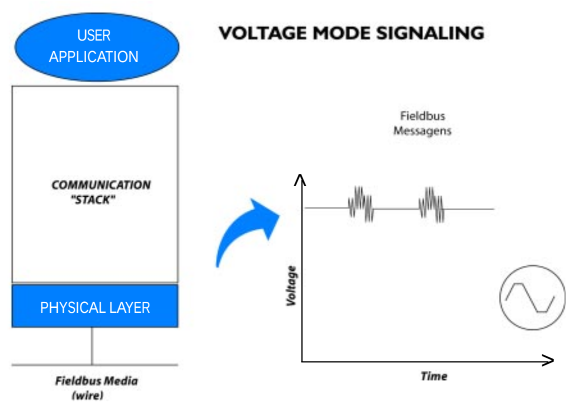
* Protocol Control Information
** Protocol Data Unit
*** There may be more than one octet of preamble if repeaters are used

Physical Layer

The Physical Layer is defined by approved standards from the International Electrotechnical Commission (IEC) and The International Society of Measurement and Control (ISA).

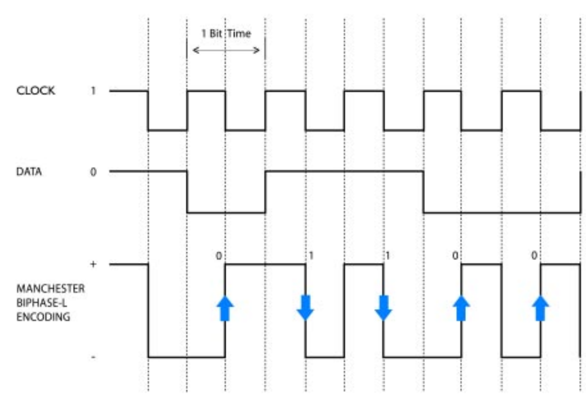
The Physical Layer receives messages from the communication stack and converts the messages into physical signals on the fieldbus transmission medium and vice-versa.

Conversion tasks include adding and removing preambles, start delimiters, and end delimiters.



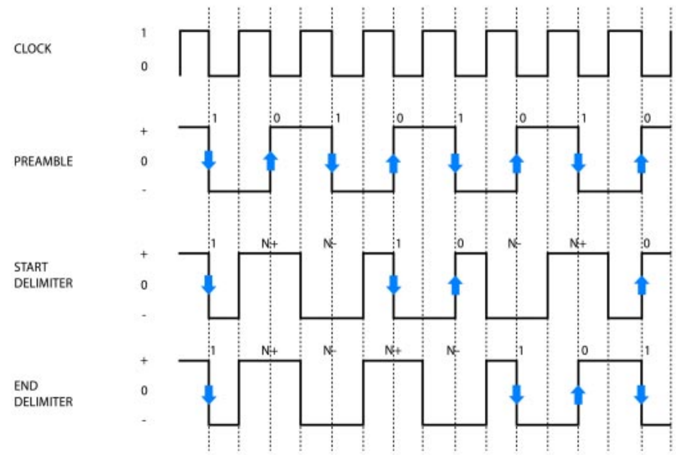
Example of Voltage mode signaling.

Fieldbus signals are encoded using the well-known Manchester Biphase-L technique. The signal is called "synchronous serial" because the clock information is embedded in the serial data stream. Data is combined with the clock signal to create the fieldbus signal as shown in the figure below. The receiver of the fieldbus signal as shown in the figure below. The receiver of the fieldbus signal interprets a positive transition in the middle of a bit time as a logical "O" and a negative transition as a logical "1".



Manchester Biphase-L Encoding.

Special characters are defined for the preamble, start delimiter, and end delimiter.



Preamble, Start Delimiter, and End Delimiter.

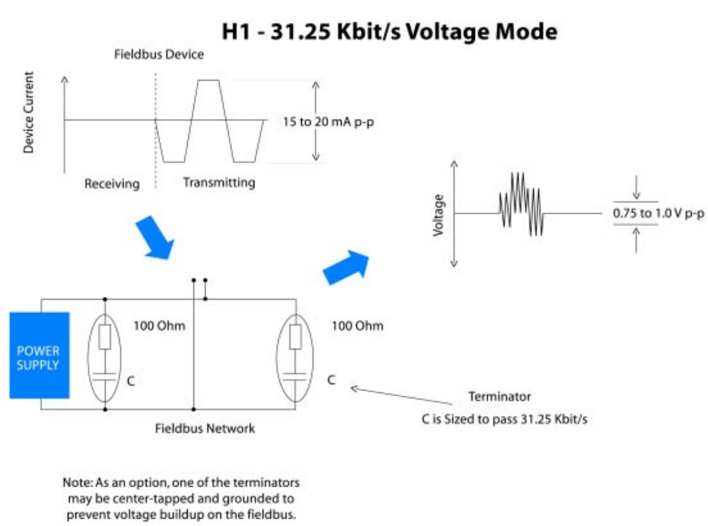
The preamble is used by the receiver to synchronize its internal clock with the incoming fieldbus signal.

Special N+ and N- codes are in the start delimiter and end delimiter. Note that the N+ and N- signals do not transition in the middle of a bit time. The receiver uses the start delimiter to find the beginning of a fieldbus message. After it finds the start delimiter, the receiver accepts data until the end delimiter is received.

31.25 kbit/s Fieldbus Signaling

The transmitting device delivers + 10 mA at 31.25 kbit/s into a 50 ohm equivalent load to create a 1.0 volt peak-to-peak voltage modulated on top of the direct current (DC) supply voltage.

The DC supply voltage can range from 9 to 32 volts, however for I.S. applications, the allowed power supply voltage depends on the barrier rating.



Fieldbus Signaling

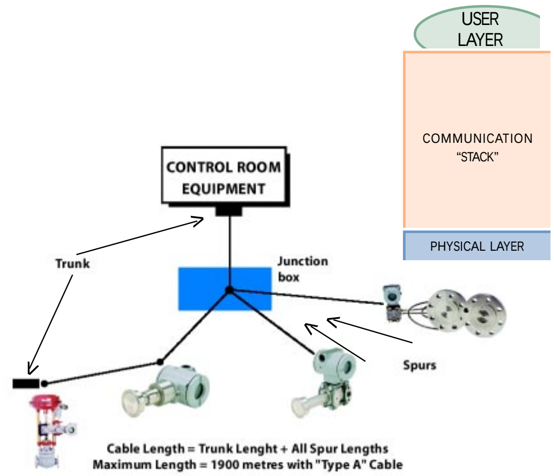
31.25 kbit/s devices can be powered directly from the fieldbus and can operate on wiring that was previously used for 4-20 mA devices.

The 31.25 kbit/s fieldbus also supports intrinsically safe (I.S.) fieldbuses with bus powered devices. To accomplish this, an I.S. barrier is placed between the power supply in the safe area and the I.S. device in the hazardous area.

31.25 kbit/s Fieldbus Wiring

Fieldbus allows stubs or "spurs".

Fieldbus Wiring

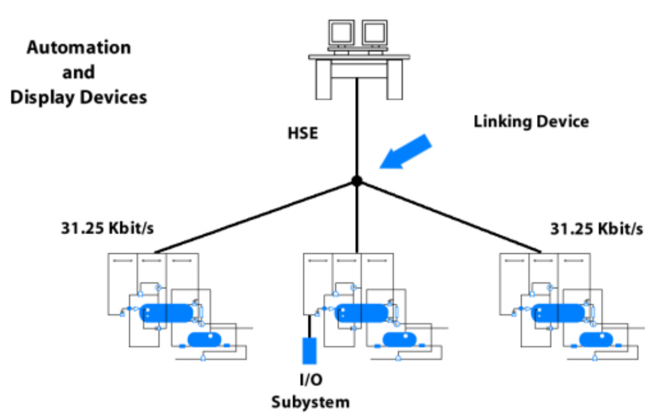


The length of the fieldbus is determined by the communication rate, cable type, wire size, bus power option, and I.S. option.

High Speed Ethernet

A **Linking Device** is used to interconnect 31.25 kbit/s fieldbuses and make them accessible to a High Speed Ethernet (HSE) backbone running at 100 Mbit/s or 1 Gbit/s. The I/O Subsystem Interface shown in the figure allows other networks such as DeviceNetâ and Profibusâ to be ampped into standard FOUDATIONTMFieldbus function blocks. The I/O Subsystem Interface can be connected to the 31.25 Kbit/s fieldbus or HSE.

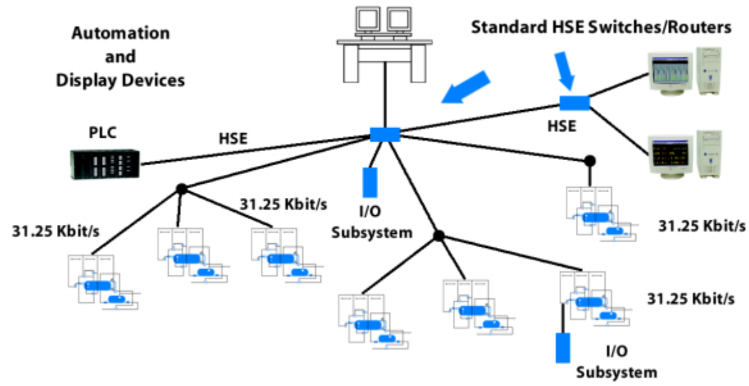
Linking Device



TUTORIAL

Since all of the 31.25 kbit/s FOUDATION™Fieldbus messages are communicated on the HSE using standard Ethernet protocols (e.g., TCP/IP, SNTP, SNMP, etc.), commercial off-the-shelf HSE equipment such as Switches and Routers are used to create larger networks. OF course all or part of the HSE network can be made redundant to achieve the level fault tolerance needed by the application.

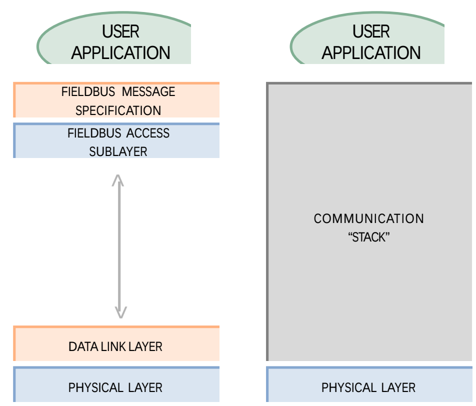
Large Networks.



Communication Stack

The following sections will describe the operation of the layers in the Communication Stack.

Communication Stack.



The Data Link Layer (DLL)

Layer 2, the Data Link Layer (DLL), controls transmission of messages onto the fieldbus. The DLL manages access to the fieldbus through a deterministic centralized bus scheduler called the Link Active Scheduler (LAS).

The DLL is a subset of the emerging IEC/ISA DLL standard.

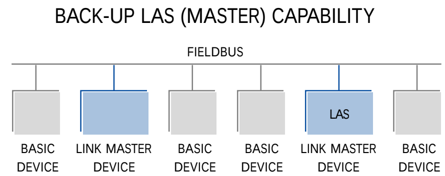
Device Types

Two types of devices are defined in the DLL specification:

- Basic Device
- Link Master

Link Master devices are capable of becoming the Link Active Scheduler (LAS). Basic devices do not have the capability to become the LAS.

Link Active Scheduler.



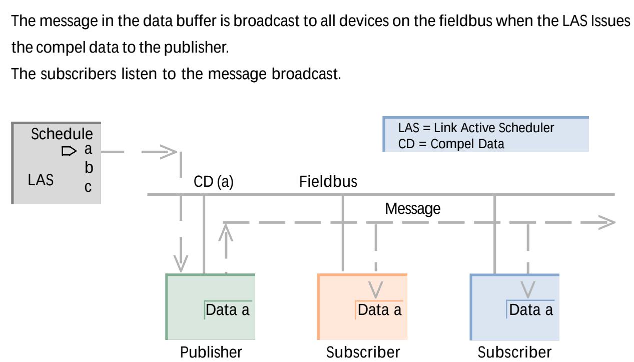
Scheduled Communication

The Link Active Scheduler (LAS) has a list of transmit times for all data buffers in all devices that need to be cyclically transmitted.

When it is time for a device to send a buffer, the LAS issues a Compel Data (CD) message to the device.

Upon receipt of the CD, the device broadcasts or "publishes" the data in the buffer to all devices on the fieldbus. Any device that is configured to receive the data is called a "subscriber".

Schedule Data Transfer.



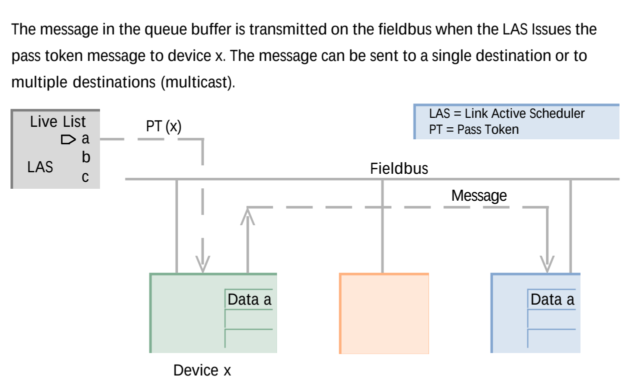
Scheduled data transfers are typically used for the regular, cyclic transfer of control loop data between devices on the fieldbus.

Unscheduled Communication

All of the devices on the fieldbus are given a chance to send "unscheduled" messages between transmissions of scheduled messages.

The LAS grants permission to a device to use the fieldbus by issuing a pass token (PT) message to the device. When the device receives the PT, it is allowed to send messages until it has finished or until the "maximum token hold time" has expired, whichever is the shorter time.

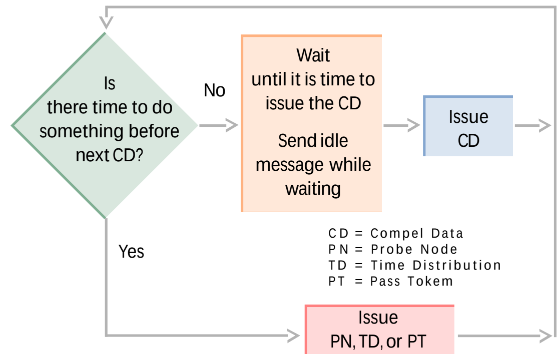
Unscheduled Data Transfer.



TUTORIAL

Link Active Scheduler Operation

The following sections describe the overall operation of the Link Active Scheduler (LAS). The algorithm used by the LAS is shown in the next Figure.



Link Active Schedule Algorithm.

CD Schedule

The CD Schedule contains a list of activities that are scheduled to occur on a cyclic basis. At precisely the scheduled time, the LAS sends a Compel Data (CD) message to a specific data buffer in a fieldbus device. The device immediately broadcasts or "publishes" a message to all devices on the fieldbus. This is the highest priority activity performed by the LAS. The remaining operations are performed between scheduled transfers.

Live List Maintenance

The list of all devices that are properly responding to the Pass Token (PT) is called the "Live List".

New devices may be added to the fieldbus at any time. The LAS periodically sends Probe Node (PN) messages to the addresses not in the Live List. If a device is present at the address and receives the PN, it immediately returns a Probe Response (PR) message. If the device answers with a PR, the LAS adds the device to the Live List and confirms its addition by sending the device a Node Activation message.

The LAS is required to probe at least one address after it has completed a cycle of sending PTs to all devices in the Live List.

The device will remain in the Live List as long as it responds properly to the PTs sent from the LAS. The LAS will remove a device from the Live List if the device does not either use the token or immediately return it to the LAS after three successive tries.

Whenever a device is added or removed from the Live List, the LAS broadcasts changes to the Live List to all devices. This allows each device to maintain a current copy of the Live List.

Data Link Time Synchronization

The LAS periodically broadcasts a Time Distribution (TD) message on the fieldbus so that all devices have exactly the same data link time. This is important because scheduled communications on the fieldbus and scheduled function block executions in the User Application are based on information obtained from these messages.

Token Passing

The LAS sends a Pass Token (PT) message to all devices in the Live List. The device is allowed to transmit unscheduled messages when it receives the PT.

LAS Redundancy

A fieldbus may have multiple Link Masters. If the current LAS fails, one of the Link Masters will become the LAS and the operation of the fieldbus will continue. The fieldbus is designed to "fail operational".

Fieldbus Access Sublayer (FAS)

The FAS uses the scheduled and unscheduled features of the Data Link Layer to provide a service for the Fieldbus Message Specification (FMS). The types of FAS services are described by Virtual Communication Relationships (VCR).

The VCR is like the speed dial feature on your memory telephone. There are many digits to dial for an international call such as international access code, country code, city code, exchange code and finally the specific telephone number.

This information only need to be entered once and then a "speed dial number" is assigned.

After setup, only the speed dial number needs to be entered for the dialing to occur. Likewise, after configuration, only the VCR number is needed to communicate with another fieldbus device.

Just as there are different types of telephone calls such as person to person, collect, or conference calls, there are different types of VCRs.

Client/Server VCR Type

The Client/Server VCR Type is used for queued, unscheduled, user initiated, one to one, communication between devices on the fieldbus.

Queued means that messages are sent and received in the order submitted for transmission, according to their priority, without overwriting previous messages.

When a device receives a Pass Token (PT) from the LAS, it may send a request message to another device on the fieldbus. The requester is called the "Client" and the device that received the request is called the "Server." The Server sends the response when it receives a PT from the LAS.

The Client/Server VCR Type is used for operator initiated requests such as setpoint changes, tuning parameter access and change, alarm acknowledge, and device upload and download.

Report Distribution VCR Type

The Report Distribution VCR Type is used for queued, unscheduled, user initiated, one to many communications.

When a device with an event or a trend report receives a Pass Token (PT) from the LAS, it sends its message to a "group address" defined for its VCR. Devices that are configured to listen on that VCR will receive the report.

The Report Distribution VCR Type is typically used by fieldbus devices to send alarm notifications to the operator consoles.

Publisher/ Subscriber VCR type

The Publisher/Subscriber VCR Type is used for buffered, one to many communications.

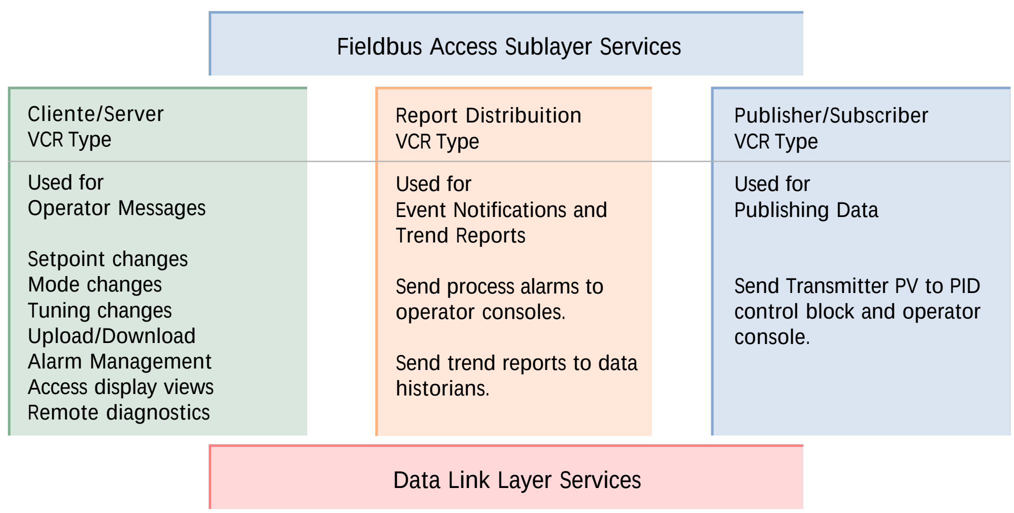
Buffered means that only the latest version of the data is maintained within the network. New data completely overwrites previous data.

When a device receives the Compel Data (CD), the device will "Publish" or broadcast its message to all devices on the fieldbus. Devices that wish to receive the Published message are called "Subscribers".

The CD may be scheduled in the LAS, or it may be sent by Subscribers on an unscheduled basis. An attribute of the VCR indicates which method is used.

The Publisher/Subscriber VCR Type is used by the field devices for cyclic, scheduled, publishing of User Application function block input and outputs such as process variable (PV) and primary output (OUT) on the fieldbus.

Summary of VCR Types



Fieldbus Message Specification (FMS)

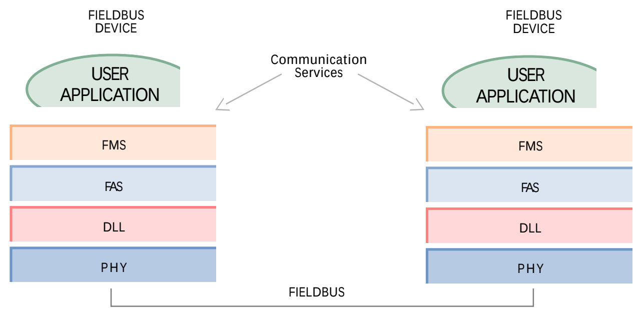
Fieldbus Message Specification (FMS) services allow user applications to send messages to each other across the fieldbus using a standard set of message formats.

FMS describes the communication services, message formats, and protocol behavior needed to build messages for the User Application.

Data that is communicated over the fieldbus is described by an "object description." Object descriptions are collected together in a structure called an "object dictionary" (OD).

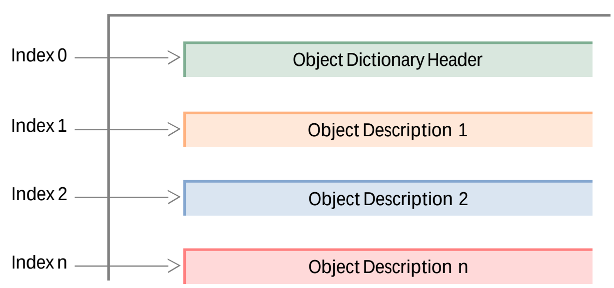
The object description is identified by its "index" in the OD. Index 0, called the object dictionary header, provides a description of the dictionary itself, and defines the first index for the object descriptions of the User Application. The User Application object descriptions can start at any index above 255.

Fieldbus Message Specification.



Index 255 and below define standard data types such as Boolean, integer, float, bitstring, and data structures that are used to build all other object descriptions.

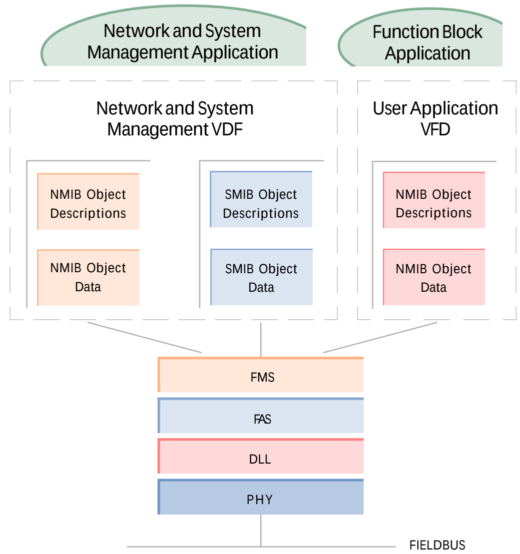
Object Dictionary.



Virtual Field Device (VFD)

A "Virtual Field Device" (VFD) is used to remotely view local device data described in the object dictionary. A typical device will have at least two VFDs.

TUTORIAL



Virtual Field Device.

Network Management is part of the Network and System Management Application. It provides for the configuration of the communication stack. The Virtual Field Device (VFD) used for Network Management is also used for System Management. This VFD provides access to the Network Management Information Base (NMIB) and to the System Management Information Base (SMIB). NMIB data includes Virtual Communication Relationships (VCR), dynamic variables, statistics, and Link Active Scheduler (LAS) schedules (if the device is a Link Master). SMIB data includes device tag and address information, and schedules for function block execution.

System Management is described further in the User Application section.

Communication Services

FMS communication services provide a standardized way for user applications such as function blocks to communicate over the fieldbus. Specific FMS communication services are defined for each object type.

All of the FMS services can only use the Client/Server VCR Type except as noted.

Context Management Services

The following FMS services are used to establish and release Virtual Communications Relationships (VCR) with, and determine the status of a VFD.

Initiate	Establish communications
Abort	Release communications
Reject	Reject improper service
Status	Read a device status
UnsolicitedStatus	Send unsolicited status
Identify	Read vendor, type and version

Object Dictionary Services

The following FMS services allow the User Application to access and change the object descriptions (OD) in a VFD.

GetOD	Read an object dictionary(OD)
InitiatePutOD	Start an OD
Load PutOD	Load an OD into a device
TerminatePutOD	Stop an OD Load

Variable Access Services

The following FMS services allow the user application to access and change variables associated with an object description.

Read	Read a variable
Write	Write a variable
InformationReport	Send Data*
DefineVariableList	Define a Variable List
DeleteVariableList	Delete a Variable List

*Can use Publisher/Subscriber or Report Distribution VCR Types.

Event Services

The following FMS services allow the user application to report events and manage event processing.

EventNotification	Report an event*
AcknowledgeEventNotification	Acknowledge an event
AlterEventConditionMonitoring	Disable / Enable event *

* Can use Report Distribution VCR Type

TUTORIAL

Upload/Download Services

It is often necessary to remotely upload or download data and programs over the fieldbus, especially for more complex devices such as programmable logic controllers.

To allow uploads and downloads using the FMS services, a "Domain" is used. A Domain represents a memory space in a device.

The following FMS services allow the User Application to upload and download a Domain in a remote device.

Program Invocation Services

The "Program Invocation" (PI) allows the execution of a program in one device to be controlled remotely.

RequestDomainUpload	Request Upload
InitiateUploadSequence	Open Upload
UploadSegment	Read data from device
TerminateUploadSequence	Stop Upload
RequestDomainDownload	Request Download
InitiateDownloadSequence	Open Download
Downloadsegment	Send data to device
TerminateDownloadSequence	Stop Download

A device could download a program into a Domain (see previous section) of another device using the download service and then remotely operate the program by issuing PI service requests.

The state diagram for the PI is shown as an example of FMS protocol behavior later in this document.

Message Formatting

The exact formatting of FMS messages is defined by a formal syntax description language called

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

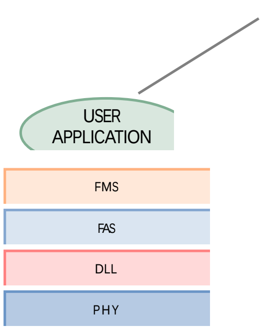
Abstract Syntax Notation 1 (ASN.I).

ASN.1 was developed by the International Telegraph and Telephone Consultative Committee (CCITT) in the early 1980s as a part of the CCITT mail standardization activities.

See next Figure for a partial example of ASN.1 definition for the FMS Read service.

Definition of a Read Request.

```
Read__Request::=SEQUENCE {
Access-specification    CHOICE {
        Index            (0) IMPLICIT Index,
        variable-name    (1) IMPLICIT Name,
        variable-list-name        (2) IMPLICIT Name,
        }.
        sub-index                 (3) IMPLICIT subindex OPTIONAL
        }
```



This example states that the items Access-specification and sub-index occur in SEQUENCE in the message.

The Access-specification is a CHOICE of using either an index or a name to access a variable.

The sub-index is OPTIONAL. It is used only to select an individual element of an array or record variable.

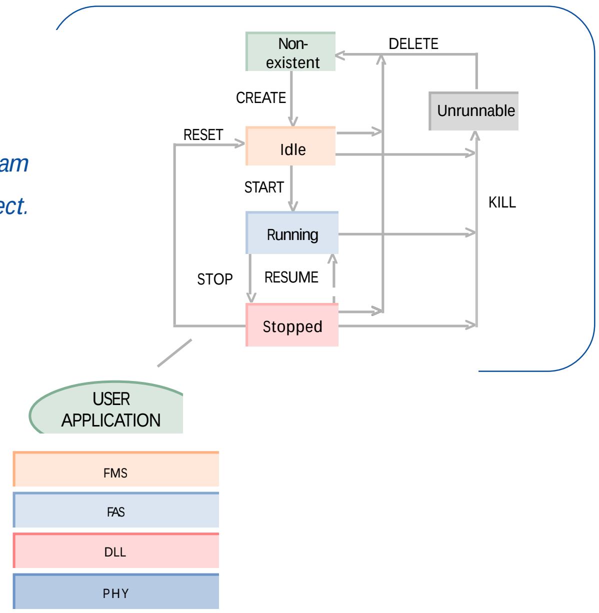
The numbers in the brackets are the actual encoding numbers that are used to identify the fields

Protocol Behavior

in an encoded message.

Certain types of objects have special behavioral rules that are described by the FMS specification. For example, the simplified behavior of a Program Invocation object is shown in the next Figure.

Behavior Rules for the Program Invocation Object.



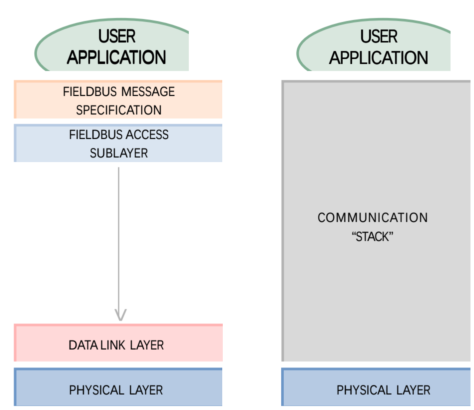
A remote device can control the state of the program in another device on the fieldbus. For example, the remote device would use the Create Program Invocation FMS service to change the program state from Non-existent to Idle.

The Start FMS service would be used to change the state from Idle to Running and so on.

User Application Blocks

The Fieldbus Foundation has defined a standard User Application based on “Blocks.” Blocks are representations of different types of application functions.

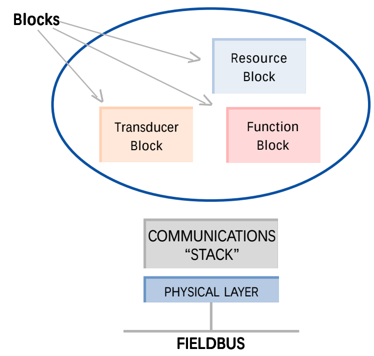
User Application Based on “Blocks.”



TUTORIAL

The types of blocks used in a User Application are described in the next Figure.

User Application - Blocks



Resource Block

The Resource Block describes characteristics of the fieldbus device such as the device name, manufacturer, and serial number. There is only one resource block in a device.

Function Block

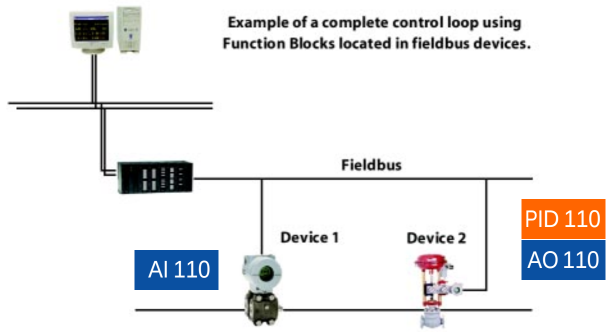
Function Blocks (FB) provide the control system behavior. The input and output parameters of Function Blocks can be linked over the fieldbus. The execution of each Function Block is precisely scheduled. There can be many function blocks in a single User Application.

The Fieldbus Foundation has defined sets of standard Function Blocks. Ten standard Function Blocks for basic control are defined by the Standard FF-891 Function Blocks — Part 2 specification. These blocks are listed below.

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

Nineteen Additional Standard Function Blocks for advanced control are defined in the Part 3 specification of this standard.

Function blocks can be built into fieldbus devices as needed to achieve the desired device functionality. For example, a simple temperature transmitter may contain an AI function block. A control valve might contain a PID function block as well as the expected AO block.



Control loop Using Function Blocks.

Thus a complete control loop can be built using only a simple transmitter and a control valve.

Transducer Blocks

Transducer Blocks decouple Function Blocks from the local input/output functions required to read sensors and command output hardware. They contain information such as calibration date and sensor type. There is usually one transducer block for each input or output function block.

The following additional objects are defined in the User Application:

Link Obiects define the links between Function Block inputs and outputs internal to the device and across the fieldbus network.

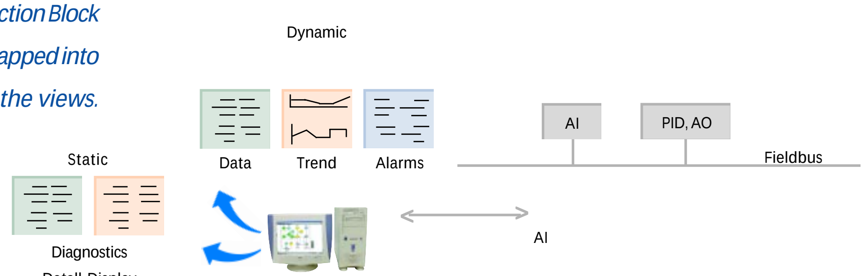
Trend Obiects allow local trending of function block parameters for access by hosts or other devices.

Alert Obiects allow reporting of alarms and events on the fieldbus.

View Obiects are predefined groupings of block parameter sets that can be used by the human/ machine interface. The function block specification defines four views for each type of block.

The next Figure shows an example of how common Function Block variables map into the views.

Function Block variables mapped into the views.



XYZ Block	View 1 Operation Dynamic	View_2 Operation Static	View_3 All Dynamic	View_4 Other Static
SP	X		X	
PV	X		X	
SP HI LIMIT		X		
CAS IN			X	
GAIN				X

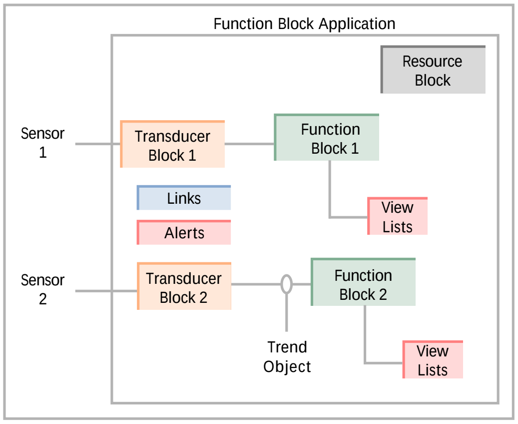
Only a partial listing of the block parameters is shown in the example.

- VIEW_1 - Operation Dynamic - Information required by a plant operator to run the process.
- VIEW_2 - Operation Static - Information which may need to be read once and then displayed along with the dynamic data.
- VIEW_3 - All Dynamic - Information which is changing and may need to be referenced in a detailed display.
- VIEW_4 - Other Static - Configuration and maintenance information.

Fieldbus Device Definition

The function of a fieldbus device is determined by the arrangement and interconnection of blocks.

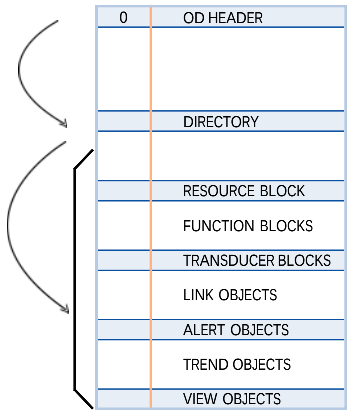
Function Block variables mapped into the views.



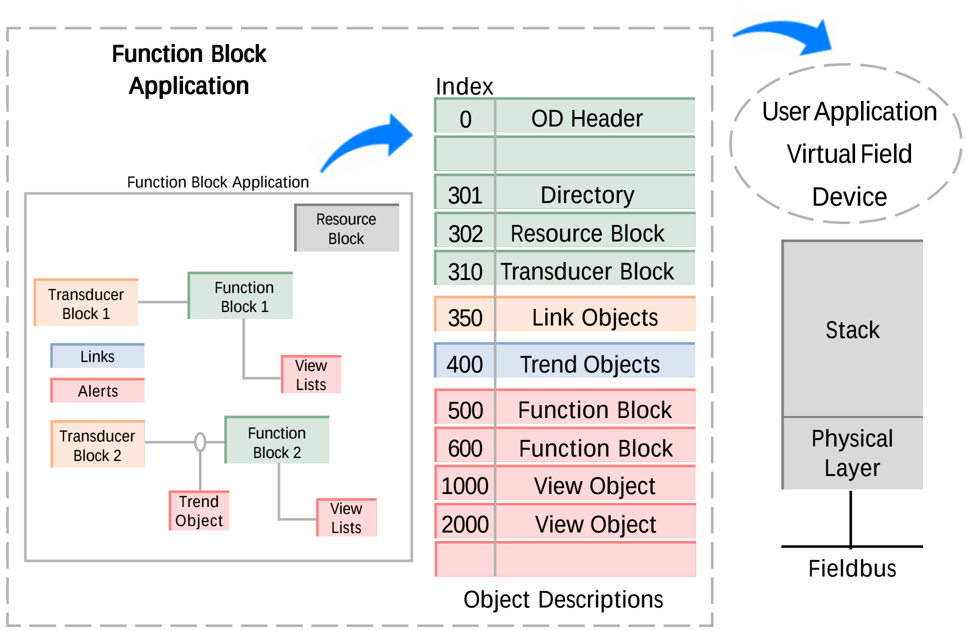
The device functions are made visible to the fieldbus communication system through the User Application Virtual Field Device (VFD) discussed earlier.

The header of the User Application object dictionary points to a Directory which is always the first entry in the function block application. The Directory provides the starting indexes of all of the other entries used in the Function Block application.

Function Block application.



The VFD object descriptions and their associated data are accessed remotely over the fieldbus network using Virtual Communication Relationships (VCRs) as shown below .



Fieldbus network using Virtual Communication Relationships.

System Management

Function Blocks must execute at precisely defined intervals and in the proper sequence for correct control system operation.

System management synchronizes execution of the Function Blocks and the communication of function block parameters on the fieldbus.

System management also handles other important system features such as publication of the time of day to all devices, including automatic switchover to a redundant time publisher, automatic assignment of device addresses, and searching for parameter names or "tags" on the fieldbus.

All of the configuration information needed by System Management such as the Function Block schedule is described by object descriptions in the Network and System Management Virtual Field Device (VFD) in each device. This VFD provides access to the System Management Information Base (SMIB), and also to the Network Management Information Base (NMIB).

Function Block Scheduling

A schedule building tool is used to generate function block and Link Active Scheduler (LAS) schedules. Assume that the schedule building tool has built the following schedules for the loop previously described in the section "Function Block."

The schedules contain the start time offset from the beginning of the "absolute link schedule start time." The absolute link schedule start time is known by all devices on the fieldbus.

Absolute Link Schedule Start Time.

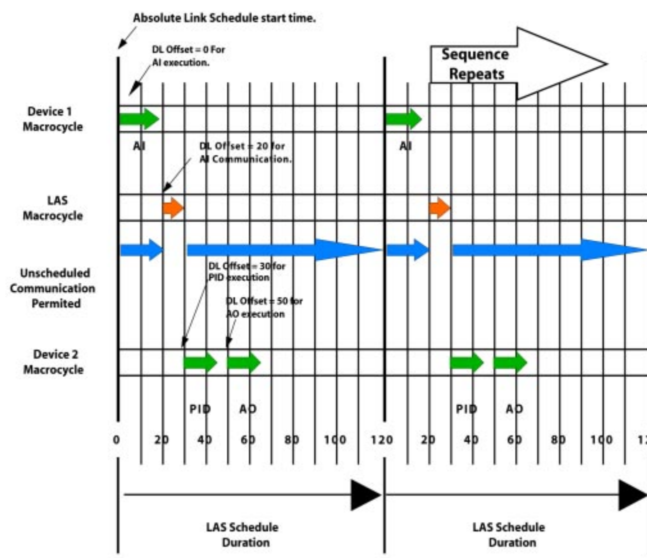
	Offset from Absolute Link Schedule Start Time
Schedule AI Function Block Extension	0
Schedule Communications of AI	20
Schedule PID Function Block Execution	30
Schedule AO Function Block Execution	50

A "macrocycle" is a single iteration of a schedule within a device. The following figure shows the relationships between the absolute link schedule start time, LAS macrocycle, device macrocycles, and the start time offsets.

In the next Figure, System Management in the transmitter will cause the AI function block to execute at offset 0. At offset 20 the Link Active Scheduler (LAS) will issue a Compel Data (CD) to the AI function block buffer in the transmitter and data in the buffer will be published on the fieldbus.

Absolute Link schedule start time.

The start of individual macrocycles is defined as an offset from the absolute link schedule start time.



At offset 30 System Management in the valve will cause the PID function block to execute followed by execution of the AO function block at offset 50.

The pattern exactly repeats itself assuring the integrity of the control loop dynamics.

Note that during the function block execution, the LAS is sending the Pass Token message to all devices so that they can transmit their unscheduled messages such as alarm notifications or operator setpoint changes.

For this example, the only time that the fieldbus can not be used for unscheduled messages is from offset 20 to offset 30 when the AI function block data is being published on the fieldbus.

Application Clock Distribution

The FOUNDATION™ Fieldbus supports an application clock distribution function. The application clock is usually set equal to the local time of day or to Universal Coordinated Time.

System Management has a time publisher, which periodically sends an application clock synchronization message to all fieldbus devices. The data link scheduling time is sampled and sent with the application clock message so that the receiving devices can adjust their local application time. Between synchronization messages, application clock time is independently maintained in each device based on its own internal clock.

Application Clock synchronization allows the devices to time stamp data throughout the fieldbus network. If there are backup application clock publishers on the fieldbus, a backup publisher will become active if the currently active time publisher should fail.

Device Address Assignment

Every fieldbus device must have a unique network address and physical device tag for the fieldbus to operate properly.

To avoid the need for address switches on the instruments, assignment of network addresses can be performed automatically by System Management.
The sequence for assigning a network address to a new device is as follows:

- A physical device tag is assigned to a new device via a configuration device. This can be done "offline" at a bench or "on-line" through special default network addresses on the fieldbus.

- Using default network addresses, System Management asks the device for its physical device tag. System Management uses the physical device tag to look up the new network address in a configuration table. System Management then sends a special "set address" message to the device which forces the device to move to the new network address.

- The sequence is repeated for all devices that enter the network at a default address.

Find Tag Service

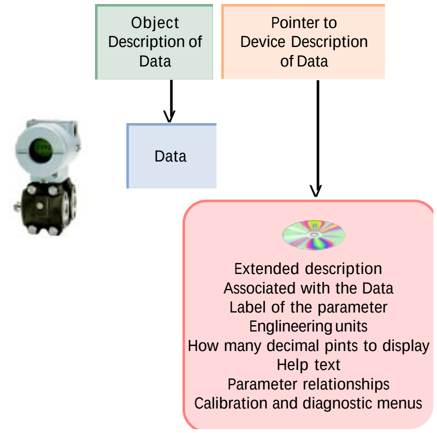
For the convenience of host systems and portable maintenance devices, System Management supports a service for finding devices or variables by a tag search.

The "find tag query" message is broadcast to all fieldbus devices. Upon receipt of the message, each device searches its Virtual Field Devices O/FD) for the requested tag and returns complete path information (if the tag is found) including the network address, VFD number, virtual communication relationship O/CR) index, and object dictionary (OD) index. Once the path is known, the host or maintenance device can access the data for the tag.

Device Descriptions

A critical characteristic required of fieldbus devices is interoperability. To achieve interoperability, Device Description (DD) technology is used in addition to standard function block parameter and behavior definitions.

The DD provides an extended description of each object in the Virtual Field Device (VFD) as shown in the next Figure.



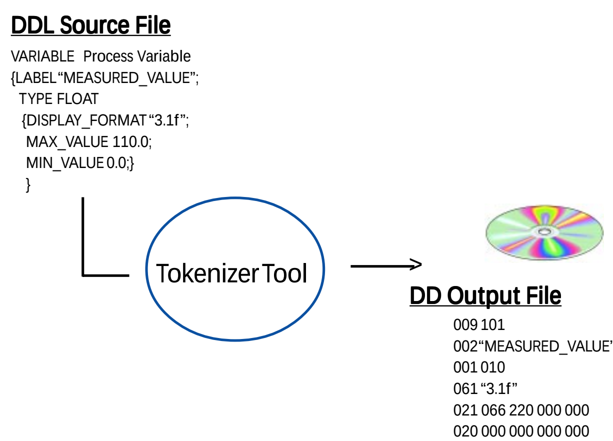
Extended description of each object in the Virtual Field Device.

The DD provides information needed for a control system or host to understand the meaning of the data in the VFD including the human interface for functions such as calibration and diagnostics. Thus the DD can be thought of as a "driver" for the device.

The DDs are similar to the drivers that your personal computer (PC) uses to operate different printers and other devices that are connected to the PC. Any control system or host can operate with the device if it has the device's DD.

Device Description Tokenizer

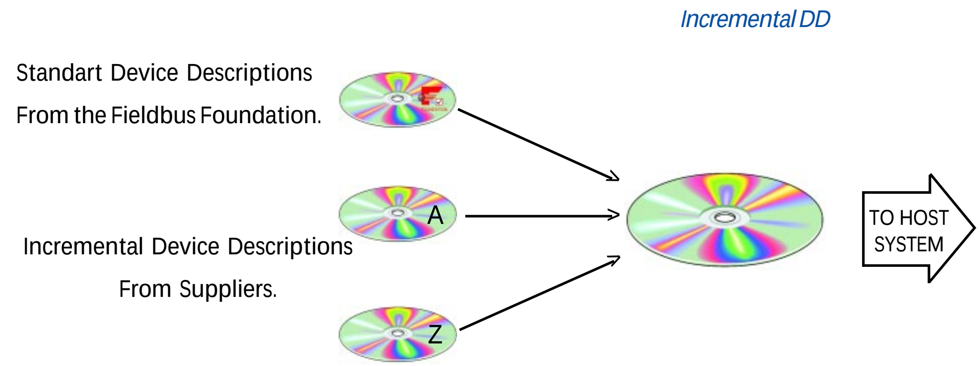
The DD is written in a standardized programming language known as Device Description Language (DDL). A PC-based tool called the "Tokenizer" converts DD source input files into DD output files by replacing key words and standard strings in the source file with fixed "tokens" as shown in next Figure.



Tokenizer.

The Fieldbus Foundation (FF) provides DDs for all standard Function Blocks and Transducer Blocks. Device suppliers will typically prepare an "incremental" DD which references the Standard DDs. Suppliers may also add supplier specific features such as calibration and diagnostic procedures to their devices. These features can also be described in the incremental DD.

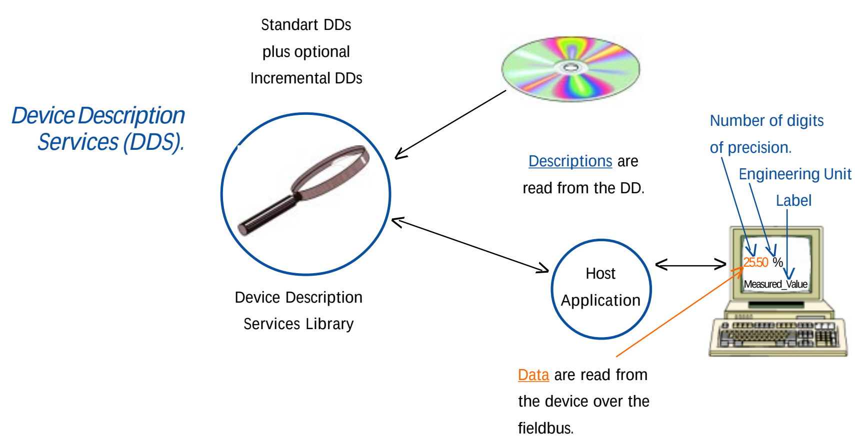
The Fieldbus Foundation makes the Standard DDs available on a CD-ROM. The user can obtain the incremental DD from the device supplier or from the Fieldbus Foundation if the supplier has registered their incremental DD with the Fieldbus Foundation.



The incremental DDs can also be read directly from the device over the fieldbus, if the device supports the upload services and contains a Virtual Field Device (VFD) for the DD.

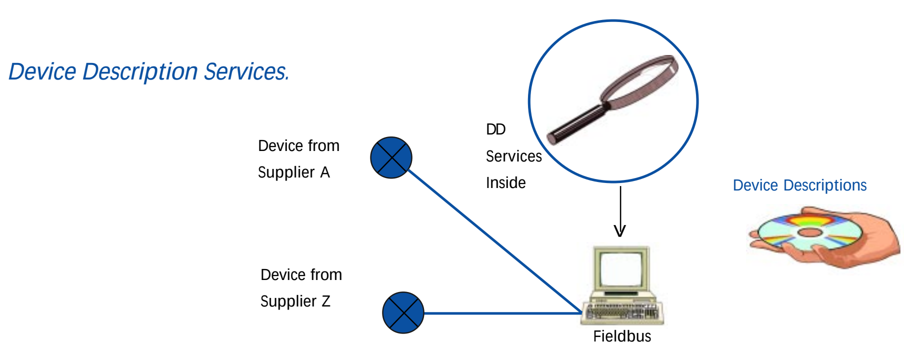
Device Description Services (DDS)

On the host side, library functions called Device Description Services (DDS) are used to read the device descriptions.



Note that DDS reads descriptions, not operational values. The operational values are read from the fieldbus device over the fieldbus using FMS communication services.

New devices are added to the fieldbus by simply connecting the device to the fieldbus wire and providing the control system or host with the standard and incremental (if any) DD for the new device.

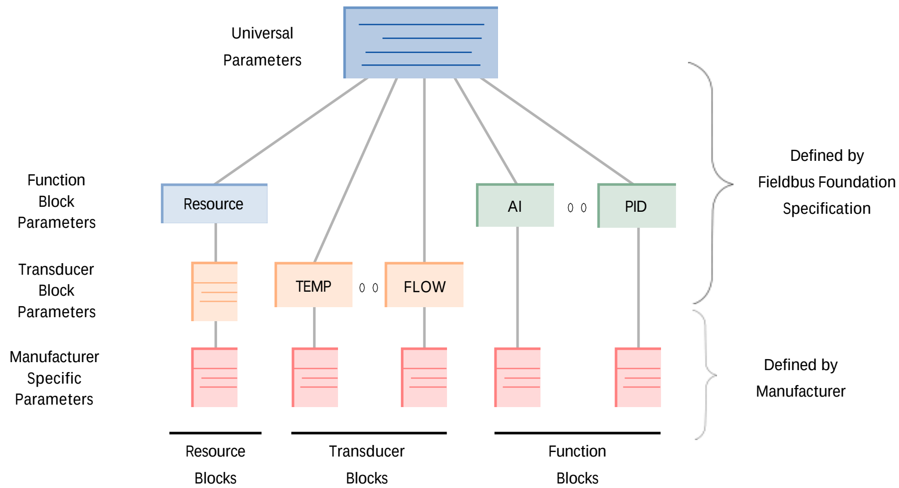


DDS technology allows operation of devices from different suppliers on the same fieldbus with only one version of the host human interface program.

Device Description Hierarchy

The Fieldbus Foundation has defined a hierarchy of Device Descriptions (DD) to make it easier to build devices and perform system configuration. The hierarchy is shown in the next Figure.

Device Description Hierarchy.



The first level in the hierarchy is the Universal Parameters. Universal Parameters consist of common attributes such as Tag, Revision, Mode, etc. All blocks must include the Universal Parameters.

The next level in the hierarchy is the Function Block Parameters. At this level, parameters are defined for the standard Function Blocks. Parameters for the standard Resource Block are also defined at this level.

The third level is called Transducer Block Parameters. At this level, parameters are defined for the standard Transducer Blocks. In some cases, the transducer block specification may add parameters to the standard Resource Block.

The Fieldbus Foundation has written the Device Descriptions for the first three layers of the hierarchy. These are the standard Fieldbus Foundation DDs.

The fourth level of the hierarchy is called Manufacturer Specific Parameters. At this level, each manufacturer is free to add additional parameters to the Function Block Parameters and Transducer Block Parameters. These new parameters will be included in the “incremental” DD discussed earlier.

Interoperability

Each manufacturer will provide the Fieldbus Foundation with an interoperability test report for each device.

The test report identifies the Universal, Function Block, Transducer Block, and Manufacturer Specific Parameters in the device. An identifier called the Manufacturer’s Identification is used to correlate the device type and revision with its Device Description and DD revision.

Any host using the Device Description Services (DDS) interpreter will be able to interoperate with all parameters that have been defined in the device by reading the device's DD.

System Configuration

Fieldbus system configuration consists of two phases:

1 – System Design;

2 – Device Configuration.

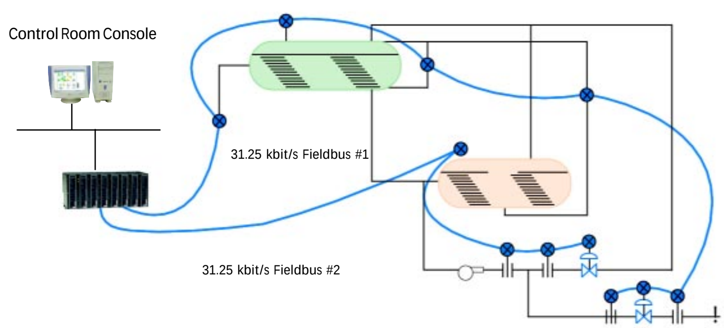
System Design

The system design for fieldbus-based systems is very similar to today's Distributed Control Systems (DCS) design with the following differences.

The first difference is in the physical wiring due to the change from 4-20 mA analog point-to-point wiring to a digital bus wiring where many devices can be connected to one wire.

Each device on the fieldbus must have a unique physical device tag and a corresponding network address.

The second difference is the ability to distribute some of the control and input/output (I/O) sub system functions from the control system to the fieldbus devices. This may reduce the number of rack mounted controllers and remote mounted I/O equipment needed for the system design.

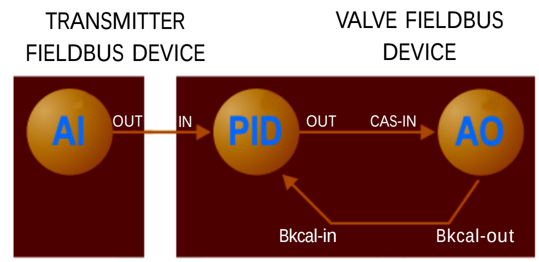


System Design.

Device Configuration

After the system design is completed and the instruments have been selected, the device configuration is performed by connecting Function Block inputs and outputs together in each device as required by the control strategy.

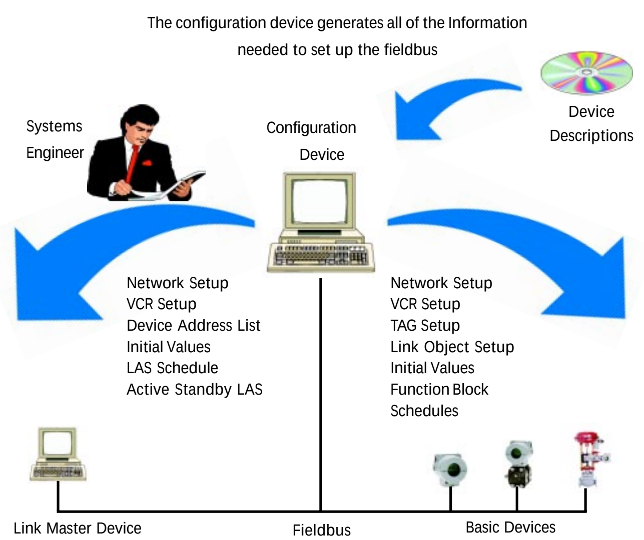
Device Configuration



After all of the function block connections and other configuration items such as device names, loop tags, and loop execution rate have been entered, the configuration device generates information for each fieldbus device.

A stand-alone loop can be configured if there is a field device that is a Link Master. This will allow continued operation of the loop without the configuration device or a central console.

Configuration Device.



The system becomes operational after the field devices have received their configurations.

Foundation Fieldbus- The right choice

The important thing that you should know is that FOUNDATION™ Fieldbus is already available to take control into your plant. It is time to make use of all its benefits and keep up to date to the latest technology. Fieldbus belongs to the time actually passing and delaying the use of it means staying behind.

Some users decide first to install small pilot units in their plants, while others go straight for a full system controlling the whole plant. There is also the possibility of purchasing fieldbus starter kits, which come with the basic apparatus (hardware and software) so the user can get acquainted on how is the process of configuring, installing and running a FOUNDATION™ Fieldbus system and feel its benefits in practical situations.

Fieldbus will prove lower investments in hardware and in installation, reduced engineering costs to configure the control strategies, powerful preventive maintenance and reporting information, which come with the increased data available from the field devices.

Do not wait for tomorrow. Call your Smar Sales Representative today and start enjoying all the fieldbus benefits, knowing that you are also saving your plant some money.