1 – Fieldbus Layers

FOUNDATION Fieldbus has several different "layers." This chapter discusses three of these layers:

1. Physical Layer: The various topologies and types of data blocks used by FOUNDATION Fieldbus
2. Communication Layer: How Fieldbus uses and assigns device registers.
3. Parameter Classes: The function or role of the information generated on the network.

This chapter provides the background on the how and what Fieldbus is. So let's start. What is Fieldbus?

Fieldbus is a bi-directional digital communication network that enables the connection of multiple field instruments and processes and operator stations (HMI). They carry out control functions and enable monitoring by means of supervision software. Figure 1-1 shows how these three layers (Field, Fieldbus, and Supervisory System) interrelate.

The FOUNDATION Fieldbus protocol was based on the ISO/OSI seven-layer communications model, although it does not include all layers. It can be divided into the Physical Layer (dealing with instrument connection techniques) and the Communication Stack (dealing with the digital communication among the devices). These are the OSI layers used by FOUNDATION Fieldbus. Figure 1-2a represents how the FOUNDATION Fieldbus maps to the OSI 7-layer model.

Figure 1-1 – Digital control system architecture

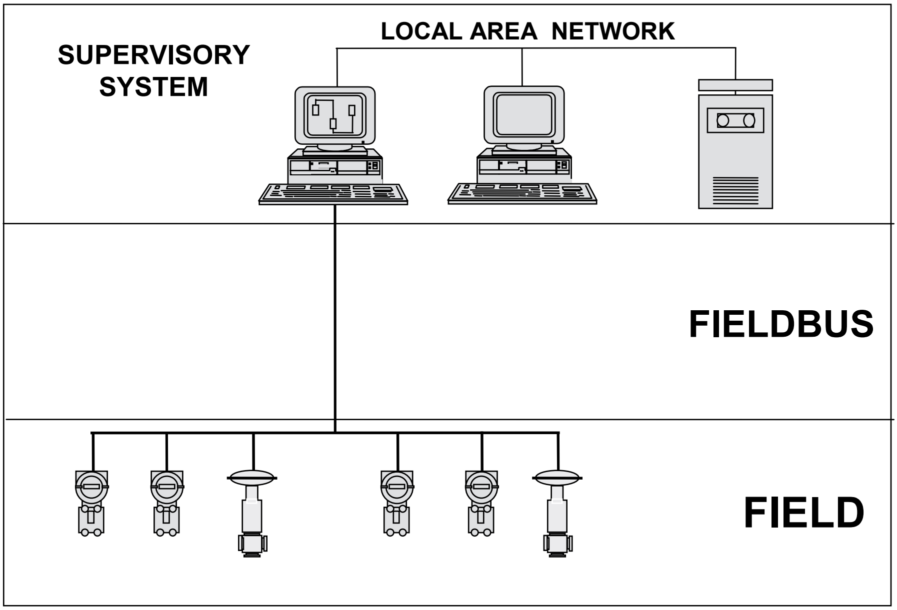
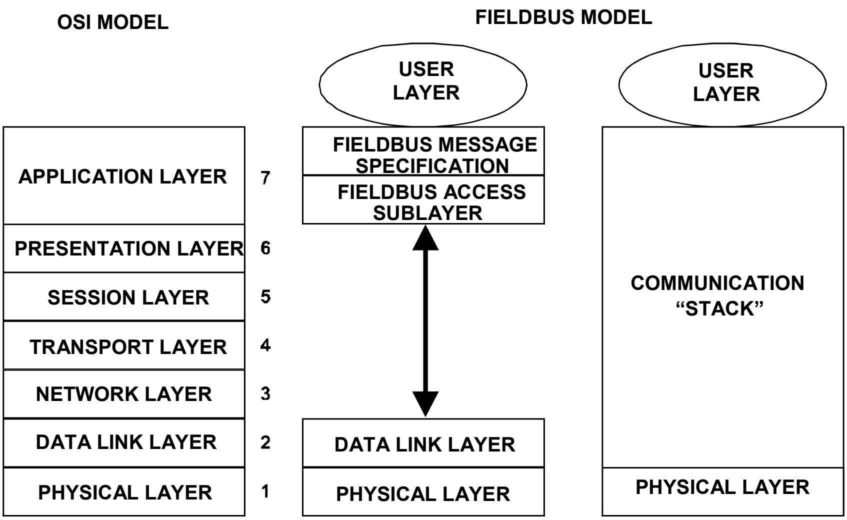


Figure 1-2a – OSI model compared with Fieldbus model



The Physical Layer is OSI layer 1, Data Link layer is OSI layer 2, and because FOUNDATION Fieldbus is a relatively simple network protocol with little cross-network communication, OSI layers 3 through 6 are not used. The Fieldbus Message Specification and Fieldbus Access Sublayer are part of OSI layer 7, and the Application Layer and the User Layer in which Function Blocks are defined reside above this. The Fieldbus Communication Stack is comprised of layers 2 through 7 of the OSI model.

As a message is transmitted from one device to another on the network, it must pass through all of the OSI layers and in the process the data packet is developed as shown in Figure 1-2b, where the numbers in the figure represent the approximate number of 8 bit octets used to transfer the user data up and down the stack.

Figure 1-2b – Fieldbus data transfer packets

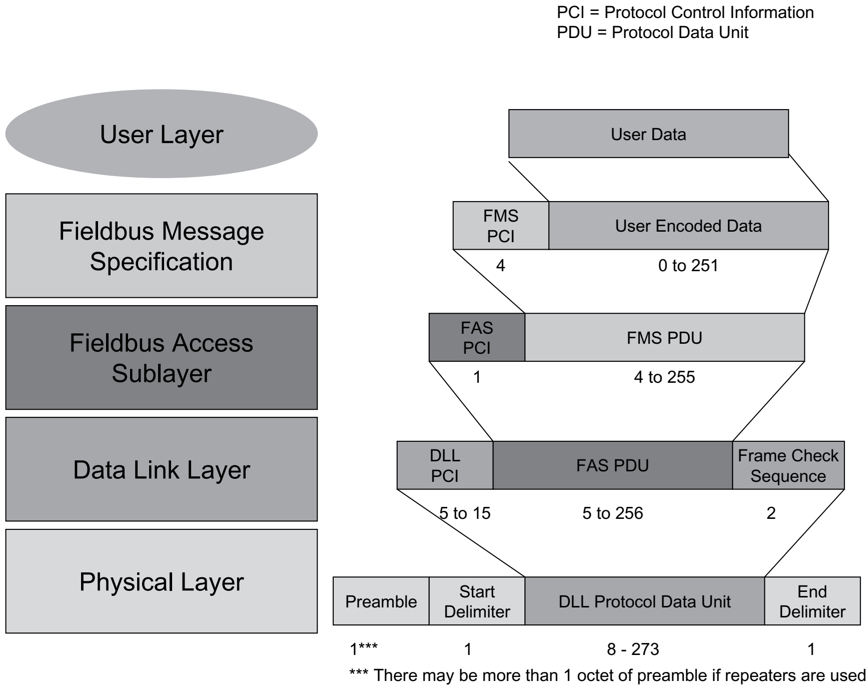
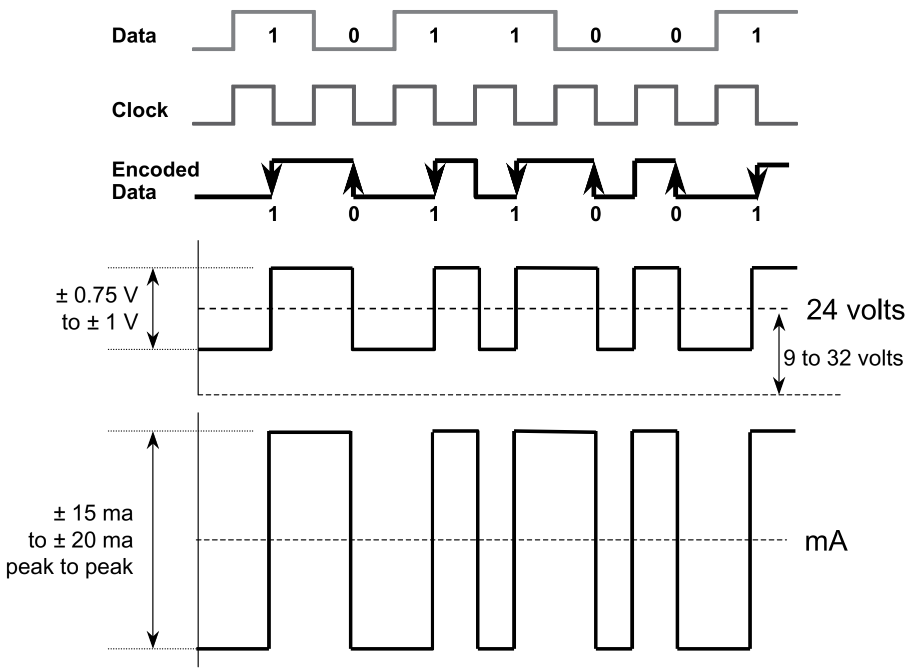


Figure 1-3 represents Manchester encoding, which is how the actual data is encoded in the H1 network. Manchester encoding adds a time reference signal to the data signal to determine the signal boundaries.

Figure 1-3 – Manchester encoding



The Data Link Layer (DLL) is a mechanism to transfer data from a node to the other nodes that need the data. The Data Link Layer also manages the priority and order of such transfer requests, as well as data, address, priority, medium control, and other parameters all related to message transfer.

Only one device on a link is allowed to use the medium (Physical Layer) at a time. The Link Active Scheduler (LAS) controls medium access.

1.1 Topology

The Application Layer provides an interface for the device's application software. This layer defines how to read, write or start a task in a remote node. The main task of this layer is to define syntax for the messages.

The main components of the Application Layer are the Fieldbus Access Sublayer (FAS) and the Fieldbus Message Specification (FMS).

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–an international access code, country code, city code, exchange code, and the specific telephone number.

This information only needs 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.

In a similar fashion, 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. VCRs and their management are covered in more detail in Chapter 5.

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.

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.

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: a Network and System Management VFD and a User Application VFD.

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 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 Schedule (LAS) schedules (if the device is a Link Master). SMIB data includes device tag and address information, and schedules for function block execution.

1.1.1 User Layer

The User Layer defines the way of accessing information within Fieldbus devices so that such information may be distributed to other devices or nodes in the Fieldbus network. This is a fundamental attribute for process-control applications.

The architecture of a Fieldbus device is based on Function Blocks, which are responsible for performing the tasks required for the current applications, such as data acquisition, feedback and cascade loop control, calculations, and actuation. Every Function Block contains an algorithm, a database (inputs and outputs), and a user-defined name (the Function Block tag number must be unique in the user's plant). Function Block parameters are addressed in the Fieldbus by means of TAG.PARAMETER-NAME.

A Fieldbus device shall include a defined quantity of Function Blocks.

Function Block. The FOUNDATION Fieldbus Function Block, especially its models and parameters–through which you can configure, maintain, and customize your applications–is a key concept of Fieldbus technology.

What is a Function Block? A Function Block is a generalized concept of the functionality in field instruments and control systems, such as analog input and output as well as PID control. The FOUNDATION Fieldbus specification "Function Block Application Process–Part 1," gives fundamental concepts, while Part 2 and later parts give various Function Blocks details.

The Function Block Virtual Field Device (VFD) contains three classes of blocks: Resource Block, Function Block, and Transducer Block.

Resource Block. A Resource Block shows what is in the VFD by providing the manufacturer's name, device name, Device Description (DD), and so on.

The Resource Block controls the overall device hardware and Function Blocks within the VFD, including hardware status.

Tip 1 – The mode of the Resource Block controls the mode of all other blocks in the device.

Transducer Block. A Function Block is a general idea while the Transducer Block is dependent on its hardware and principles of measurement. For example, a pressure transmitter and magnetic flow meter have different measurement principles but provide an analog measured value. The common part is modeled as an AI (Analog Input) Block. The difference is modeled as Transducer Blocks, which provide the information on the measurement principle. A Transducer Block is linked to a Function Block through the CHANNEL parameter of the Function Block.

In addition to converting the signal from physical to digital, Transducer Blocks are becoming ever more important because they are also the blocks used to capture and store all the diagnostic and maintenance related data for a device. A number of Standard Transducer Blocks have been defined since the second edition and these are being released in ITK version 5 and later. The initial blocks that have been standardized are temperature and pressure while the standard for partial stroke testing of valves is imminent. The Transducer Block for flow will follow.

It is end user demand and economics that are driving the need for Standard Transducer Blocks since without a standard interface to the maintenance data contained within each device, it is a cumbersome task to take full advantage of the diagnostic capabilities of a digital transmitter using modern software and asset management systems.

Transducer specifications are generally defined by the device developers. The transducer specifications establish the base scope of transducer functions. A device may have additional functions, but it **must** contain the functions specified in the specification to be interoperable within the given specification.

Function Block. A Function Block is a generalized model of measurement and control. The three Function Block classes are:

1. Standard Block, as specified by the Fieldbus Foundation.
2. Enhanced Block, with additional parameters and algorithm.
3. Open Block, Extended Block, or a Vendor-specific Block, designed by individual vendors.

The Function Blocks MAI, MAO, MDI, MDO, and FFB defined in Parts 4 and 5 of the Function Block Application Process specifications, were developed as part of the High Speed Ethernet (HSE) process. The "M-series" of blocks are able to transfer a group of eight PV signals as a single message on the Fieldbus Network and because HSE is fully backwards compatible with H1, a number of H1 devices, such as temperature multiplexers, are taking advantage of the MAI block.

The most novel of the new blocks, however, is the Fully Flexible Function Block (FFB), as it is able to be fully programmed by the end user using any of the IEC 61131 programming languages.

Like all object-based fieldbus function blocks, the FFB is a "wrapper" for the actual functions that reside and execute inside of it. The fieldbus specifications define a set of parameters that must be common to all function blocks to ensure interoperability and communications between the various blocks, devices and host system. Since each component of the fieldbus specification is treated as an object and is, to some extent, similar to a subroutine or function call in a computer program, it is possible for each manufacturer to write its own code for the object to execute as long as the results are presented in the predefined format. It is this lack of definition for the function itself that makes the FFB possible.

The FFB can be configured by the end-user with any of the IEC 61131-1 languages to whichever function is required. Thus, a device supporting the FFB can be configured or programmed for a variety of purposes from protocol converter to a nano-PLC that performs batch/recipe operations or complex multivariate control calculations such as artificial neural networks or fuzzy logic.

The FFB specification contains many useful function blocks; however, the one developed to help fieldbus in the manufacturing industry, where discrete control is more prevalent, is the device controller, which is intended to control any two- or three-state physical device. The device controller accepts a set point and causes the device to drive to that set point. Time is allowed for the transition, but alarms are generated if the physical device fails to reach the desired state or loses that state after the transition is complete. The DC block has inputs for control of the set point by external logic or commands from a host, as well as permissive, interlock, and shutdown logic functions. An operator may temporarily bypass a faulty limit switch after visual confirmation of the state of the physical device. The parameter "DC_STATE" displays one of fourteen states that describe the current control condition, while the parameter "FAIL" gives specific reasons for failures.

Unfortunately, the interfaces to program FFB are not yet fully interoperable. This means that an FFB from Manufacturer A must be programmed and configured by the host and software tools of Manufacturer B, and vice versa. However, once the FFB has been prepared and compiled through DD Services (the binary file that is used by field devices and hosts to execute the information from the DD file), it can be executed by any system supporting the FFB block type.

FFB technology was successfully demonstrated at the International Specialty Products facility in Lima, Ohio in May 2005. The demonstration consisted of converting one of the three filter beds in the process from control in the host to field-based control using linking devices containing FFBs from two manufacturers. The first FFB controlled the ten quick opening/closing valves (250 milliseconds) on one side of the filter and then control was transferred to the second linking device and its FFB to control the second bank of ten valves. After that, control was passed back to the host to control the third filter bed's operation.

Figure 1-4, "Device Description Hierarchy," shows not only how the various function blocks work together but also the different parameters that are used in each of the Standard, Enhanced, and Extended blocks available in a device. Simplistically, the Universal parameters define the basis for the Standard blocks, Enhanced Blocks build on this concept, and then manufacturers can further expand on the Enhanced blocks with their own enhancements.

Tip 2 – The function block extensions provided by manufacturers are not defined by the Foundation so may not be the same between two different manufacturers.

Despite the fact that the enhancements are not defined by the Fieldbus Foundation, they will be supported by all host systems capable of reading the associated DD and Capabilities files.

A block has a series of parameters, which have continuous indexes.