Fieldbus 104

Fieldbus blocks

- Overview
- What is a resource block?
- What is a transducer block?
- What is a function block?
- FOUNDATION fieldbus function blocks
- Basic FOUNDATION fieldbus function blocks
- Advanced FOUNDATION fieldbus function blocks
- How do function blocks get into devices?
- Instantiating blocks into devices
- Device descriptions

Overview

What are fieldbus blocks? And what do they do for me?

Think of fieldbus blocks as small, sealed software modules. Each block has a defined set of inputs and/or outputs for a specific function or type of information. You don't have to manage the internal processing that turns the inputs into outputs. That's up to the manufacturer who provides the block as part of a fieldbus device or host system.

FOUNDATION fieldbus uses three types of blocks:

- Resource blocks
- Transducer blocks

- Function blocks

Why they matter. Resource and transducer blocks provide valuable information about devices, sensors and actuators, and their performance. Function blocks are the engines of open, interoperable, device-independent control. Together, these three types of blocks make it easier for you to improve equipment performance and process control.

This course introduces you to the three types of blocks and how they're used.

Hint: As you go through the topics in this course, watch for answers to these questions:

- *What kinds of information does a resource block provide? A transducer block?*
- *What key capability do function blocks provide? Do all function blocks work the same way?*
- *Who decides how many blocks of each type are used in a particular device?*

What is a resource block?

The resource block deals with the overall device. It contains information such as manufacturer, device type, and serial number. Each device has one resource block.

In addition, the resource block also often provides information about the health or status of the device as a whole. Access to this additional information may be one of the most important features of FOUNDATION fieldbus because it can enable you to detect potential device problems before they affect the process.

During project execution, the resource block is used to identify a device, tag it, and commission it. During ongoing operations, it's used by maintenance technicians to obtain overall device configuration and status information, and to run some types of device-specific diagnostics.

There is one resource block for each device.

What is a transducer block?

The transducer block deals with the "wetted parts" of a device. It provides the local input/output functions needed to read sensors and to command actuators, displays, or other output hardware. It's the link between the physical world of sensors and actuators and the "data world" of process control.

The transducer block contains information such as calibration data, sensor type, materials of construction, and in many cases the health and operating status of actuators and sensors.

The PlantWeb advantage

In the FIELDVUE® digital valve controllers used in PlantWeb architecture, for example, the transducer block provides information that's used in valve-signature diagnostics and in checking for performance problems such as stiction. Similarly, the transducer block in a transmitter can be used to check for a variety of sensor failure conditions.

Special transducer blocks are also used to provide statistical process monitoring, predict sensor life, detect plugged impulse legs, and similar functions.

During project execution, transducer blocks are used for calibrating the device, setting units, and other tasks related to providing an accurate and reliable input or output. During ongoing operations, maintenance technicians use these blocks to troubleshoot and calibrate devices, to perform diagnostic checks, and to carry out other tasks to maintain device health and performance.

There may be several transducer blocks in a single device. For example, one transducer block may deal with the sensor or actuator, another with a local display, and a third with diagnostics.

The PlantWeb advantage

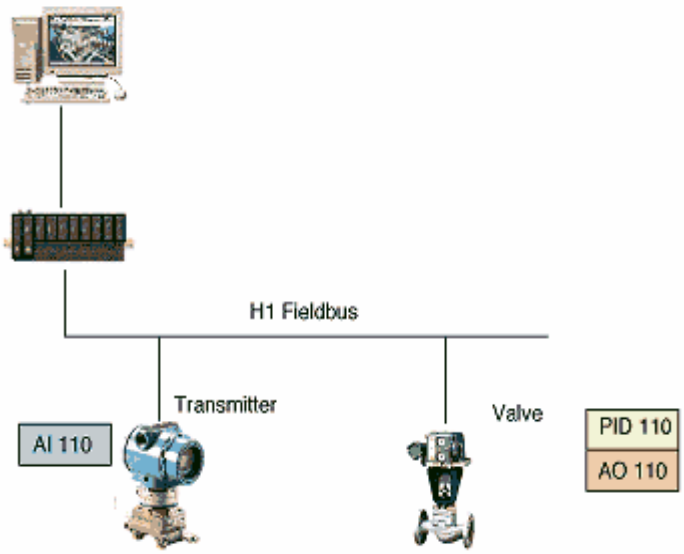
PlantWeb architecture is designed to take advantage of the tremendous diagnostic potential of resource and transducer blocks — from full 0-to-100%-stroke valve diagnostics, to sensor life and performance diagnostics, to diagnostics of external equipment like impulse legs.

Maintenance functions can tap this wealth of information through AMS Suite: Intelligent Device Manager software, and operators can see the equipment status right on the operations display.

What is a function block?

Function blocks provide control-system behavior within the fieldbus environment. Analog and discrete input and output blocks, and a wide variety of control algorithms such as characterizer, splitter, or PID, can be linked across the fieldbus to perform process control.

It's even possible -- in many cases, advantageous -- to run a control loop completely in field devices, without involving the host system.



A simple device may have only a single input or output function block. More-complex devices may have several input and output blocks, as well as blocks for monitoring and control.

During project execution, control engineers use function blocks to implement the control strategy. During ongoing operations, the function blocks provide the process-control information and functions the operators use to run the plant.

The PlantWeb advantage

PlantWeb was designed from the beginning to use fieldbus function blocks throughout the architecture -- in devices as well as the host DeltaV system.

You use the same tools to configure all function blocks, no matter where they run. You can move blocks from one device to another (or to the host) with no change in the way they run.

You can also combine analog and fieldbus devices in a single loop, or even assign a configuration designed for traditional analog installation to run in fieldbus devices instead.

The benefits: unprecedented flexibility and tremendous savings in engineering time -- with fewer errors as there is no manual mapping of function blocks between field devices and the DeltaV system.

FOUNDATION Fieldbus function blocks

The Fieldbus Foundation has defined standard sets of basic and advanced function blocks. Manufacturers decide which of these standard blocks -- as well as other, non-standard blocks -- will be supported in each fieldbus device.

Interoperability built in. If a device conforms to the FOUNDATION fieldbus specification for a function block, that function block will be interoperable -- that is, the inputs and outputs will enable the block to work together with other blocks according to the FOUNDATION fieldbus specification regardless of device or host manufacturer.

Exactly how those inputs are converted to outputs is **not** defined by the specification. Each manufacturer can choose the algorithm they use to make that happen. So while standard function blocks will work together, they may not work exactly the same.

Room for innovation. Device and system manufacturers can also provide blocks or functions that aren't specified by the FOUNDATION. For example, a manufacturer might provide a fuzzy logic block (which isn't currently defined by the fieldbus specifications), or add an auto-tuning feature to a standard PID block.

These custom blocks are interoperable with standard function blocks. That's because the FOUNDATION fieldbus specification defines inputs, outputs, modes, and other parameters even for custom blocks.

Basic FOUNDATION fieldbus function blocks

The Foundation has defined specifications for the following basic functions:

Basic Specified Continuous Blocks		
Analog Input	AI	Reads analog input
Analog Output	AO	Sends analog output
Bias Gain	B	Scaling
Control Selector	CS	Override control
Manual Loader	ML	Manual Control
PID Control	PID	PID Control
PD Control	PD	PD only control
Ratio Control	RA	Ratio Control
Basic Specified Discrete Blocks		
Discrete Input	DI	Reads discrete input
Discrete Output	DO	Sends discrete output

Advanced FOUNDATION fieldbus function blocks

The Foundation has defined specifications for the following advanced functions:

Advanced Specified Continuous Function Blocks	
Complex AO	Provides extensive interlocking
Splitter	1-in-3-out + logic -- for split ranging
Selector	4-in-1-out (min., max., mid., avg.)
Setpoint Generator	SP generator for Batch applications
Characterizer	Has interpolation and tracking
Integrator	Integrate flow or pulse + reset
Calc_A	1131-C inst. - 50 steps - analogs
Lead/Lag	Dynamic compensation
Dead Time	Delay for analog feedforward control
Analog Alarm	Provides alarm response
Advanced Specified Discrete Function Blocks	
Digital HMI	Operator input - reference by tag
Pulse Input	Pass pulses to integrator
Timer	Count up/down, debounce
Digital Alarm	Provides alarm response
Step Control	SP control using discrete actuators
Calc_D	1131-C inst. - 50 steps - discretes
Complex DO	Provides extensive interlocking
Device	Simple 2 or 3 state devices (pumps)
Dead Time	Delay for analog feedforward control
Analog Alarm	Provides alarm response

The Fieldbus Foundation adds new block type specifications on an ongoing basis. See the Fieldbus Foundation web site for the most current list.

How do function blocks get into devices?

Fieldbus function blocks enable field devices to contain control capabilities that were previously restricted to a central control system.

As a result, fieldbus-based architectures can scale up more smoothly than traditional architectures. That's because it's usually easier and cheaper to add more devices than to expand the central control system. You may not even have to add more devices, if those already installed can accommodate the functions you want to add.

But how do those functions get into the devices?

One way is for the manufacturer to include a fixed set of function blocks for each device. While some blocks can be expected in a specific device type -- such as an AI block in a transmitter, or an AO block in a valve controller -- the number and type of blocks supported will vary from device to device and vendor to vendor. One may include PID in a pressure transmitter, while another may not.

The other way is to let the user decide -- within manufacturer-set limits -- which types and how many function blocks will reside in a device. This approach is called **instantiation.**

Instantiating blocks into devices

Instantiation is the process of creating a new **instance (**or copy) of a function block in a device. It's an easy, affordable way of adding capabilities without increasing the number of devices.

For example, a pressure transmitter may be supplied with one AI block for its primary process variable, pressure. But suppose the pressure-sensor module also includes a temperature sensor to detect freezing or overheating conditions that can cause device failure. If the transmitter supports instantiation, you can instantiate a second AI block in the transmitter to monitor this temperature measurement.

Instantiation can be used for any type of function block. If you're doing cascade control, for example, you could add a second PID block to a valve controller.

Instantiation isn't possible in every situation:

- The manufacturer must design the device to allow instantiation. Some do, and some don't.

- The device must have enough available memory and processing power to support added blocks

- The device must support the specific type of function block to be added. If a transmitter only supports AI blocks, you can instantiate multiple blocks of that type — but not PID or other block types.

Device descriptions

Standard fieldbus blocks provide the basis for interoperability. But what happens if the Fieldbus Foundation adds to the list of standard function blocks, or if a manufacturer wants to offer a proprietary block or extend the capabilities of a standard block? How can existing host systems recognize the new data and capabilities provided by these blocks -- without reprogramming or upgrading the system?

The answer is a FOUNDATION fieldbus **device description** -- a software file that provides the information a host system needs to understand both the meaning of the data and the capabilities of the device. It serves as a "driver" for a device, much as a printer driver tells your PC how to access the capabilities of your printer.

For example, a device description might provide (among other things):

- The label for a parameter
- Engineering units
- Help text
- Diagnostic menus
- **Methods** for device-related tasks such as calibration

A **method** is a predefined sequence of operations executed in a device. This predefined sequence is typically used to simplify device operations such as setup and commissioning, configuration, calibration, or diagnostics. For example, a setup method in a valve instrument can guide a technician through the proper entries and options to optimize performance of the valve/actuator combination. Some devices use methods, some do not.

The Fieldbus Foundation provides device descriptions for all standard function blocks and transducer blocks, as well as distributing device descriptions that manufacturers have provided as part of the device interoperability-certification process. This ensures that users have a way to access the full capability of a device, independent of any vendor.