How to Develop Your First FOUNDATION™ Fieldbus Device

Sheldon G. Lloyd
President
Fieldbus Inc.
9390 Research Blvd. S. I-350
Austin, Texas 78759

Thomas A. Boyd
Senior Software Engineer
Fieldbus Inc.
9390 Research Blvd. S. I-350
Austin, Texas 78759

KEYWORDS

Fieldbus, Foundation Fieldbus, Interoperability, Function Block, Distributed Control, Network, Open Systems, Communications.

ABSTRACT

Foundation fieldbus has moved much of the control functionality once found only in the controller of a DCS into the field. This creates some new problems for device developers. If the developers don't have experience migrating a 4-20 mA device or a smart device to Foundation fieldbus, a lot of time can be spent just learning what to do. This paper will help guide the first time fieldbus developer through the device development process.

INTRODUCTION

This paper is directed at engineers, and engineering and marketing managers who will be responsible for development of their company's first FOUNDATION™ fieldbus product. While the concepts are not limited to the fluid process control industry, the examples and discussion are focused on this application area. The discussion is further limited to field devices such as valves and transmitters, as opposed to host systems.

We assume that readers have a thorough understanding of their own company's field device products; and a modest knowledge of Foundation fieldbus technology, which can be better learned from other sources[1].

We will describe the hardware and software components that are unique to FOUNDATION fieldbus and illustrate some typical device architectures. We will discuss the planning that should be done up-front, the essential make/buy decisions to be made, some pitfalls to be avoided, and the tools and design process required for a typical project.

CHARACTERIZATION OF A FOUNDATION FIELDBUS DEVICE

A very defining characterization of a fieldbus[2] device is that it is a product which has been registered with the Fieldbus Foundation as having passed all tests required for interoperability registration. While true, this is not very enlightening to those new to the technology. The schematics in Figures 1 and 2 illustrate the essential components that comprise a fieldbus field device. There may be additional items and different architectures, but these illustrate a minimum structure.

[1] Visit the Fieldbus Foundation's web site (www.fieldbus.org) for information on training courses and educational publications.
[2] In this paper we shall restrict the meaning of *fieldbus* to only *Foundation fieldbus* devices.

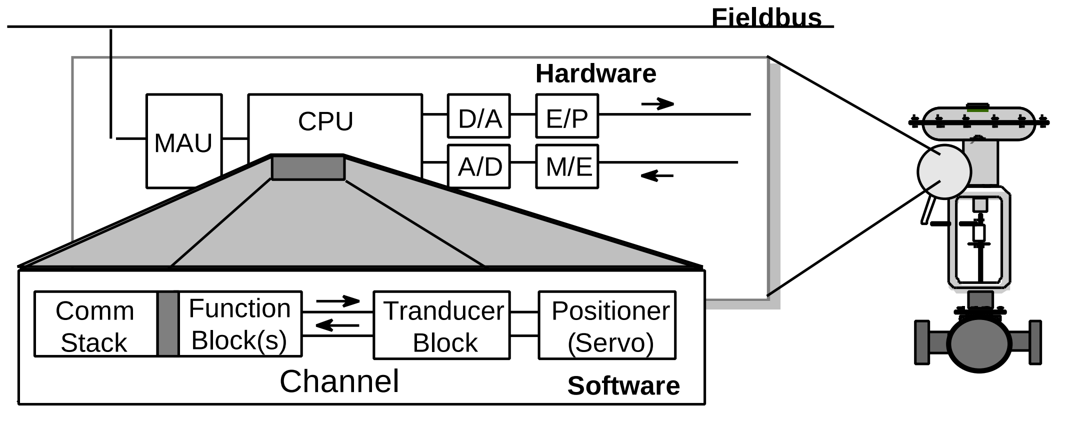

Figure 1. Basic Components of a Fieldbus Control Valve.

Figure 1 illustrates the key hardware components in the background, and software is shown as a projection from the CPU's memory. Considering the hardware first, MAU stands for Medium Attachment Unit and comprises the circuit needed to connect the device to the actual bus wires. The device will have a microprocessor, indicated here by the shorthand, CPU.

The electrical output of the microprocessor is passed through a digital to analog converter (D/A) and an electric to pneumatic converter (E/P), which results in a pressure signal that modulates the actuator. The actuator position is fed back through a mechanical to electric transducer (M/E) and an analog to digital converter (A/D), to get actual valve position back to the processor.

The software diagram illustrates four major components. The one on the far right performs the servo calculation, or positioner function for the actuator. The servo software must communicate with the D/A and A/D converters. The software and hardware for this function will be unique to the manufacturer and at this point, has nothing to do with fieldbus.

The next block to the left, the transducer block (TB), is the interface between the positioner and fieldbus. Most manufacturer specific features of the device will be done in the TB, including calibration and diagnostics. It is likely that the designer will want the TB to have access to the actual valve position. Also, any potential device failures that can be sensed should be communicated to the TB for use in diagnostics and fault detection. Although the TB is unique to a specific manufacturer's design, the data communicated to the Analog Output (AO) function block, must conform to the Foundation Fieldbus specifications [3].

The AO block is defined in detail by the Function Block Specifications. The block communicates with the TB through a numbered *channel* or channels. The AO block will not be described in depth here, but a few of its features will be highlighted. It has 30 standard (mandatory) parameters that provide it with substantial capability. It provides high and low limits on the input signal, and limits on the increasing and decreasing rates of change. There is a *fault-state* mechanism for determining the control action to be taken in the event of a failure, and a *readback* parameter that can be used by an upstream control block to avoid wind-up during limit conditions. The AO block on this valve (and the input block in a transmitter) contain a simulate parameter that allows alarms and failure procedures to be tested while the devices are connected and operating, but not during operation of the process.

[3] Fieldbus Foundation Specifications, Part 1 FF-890, Rev 1.3, and Part 2 FF-891, Rev 1.3.

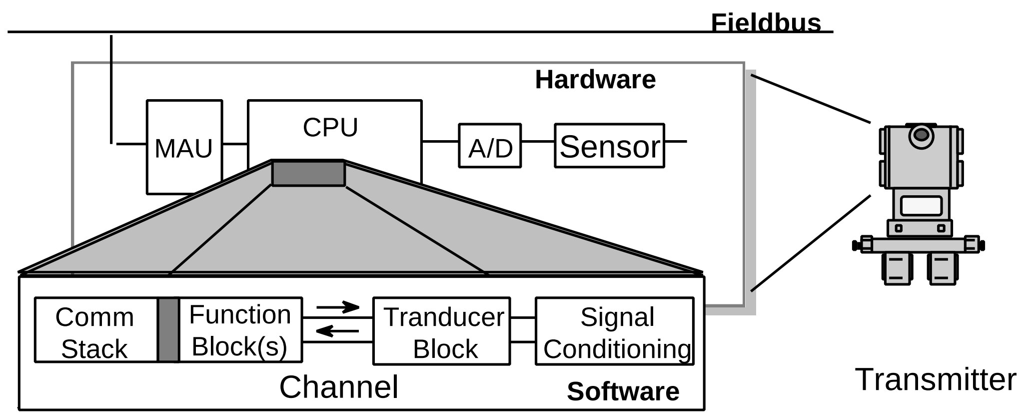

Figure 2. Basic Components of a Fieldbus Transmitter.

Figure 2 illustrates a similar architecture for a transmitter. In this case the TB interfaces with the signal conditioning circuit of the sensing device, which is manufacturer specific. Then, via a numbered channel, connects to an Analog Input (AI) function block. The AI block has 36 standard parameters. It provides a scaling function, filtering, and process and block alarms. The simulate parameter is provided, as in the AO block, for test purposes.

SIMILAR TO TRADITIONAL DEVICE DESIGN

All device development projects have two general aspects; design of the device and design of the project. Project design is common to all projects and, unfortunately, often overlooked as a critical early phase. Good engineering practice dictates that it is essential to identify all tasks required in a project and to organize these in a logical sequence. Time estimates and resource requirements for every task should be identified and documented. A project management plan, requirements document, design specification and test plans should all be part of the initial planning. For software projects we recommend a formal process based on the IEEE Software Engineering Standards.

DIFFERENT THAN TRADITIONAL DEVICE DESIGN

The differences in designing a fieldbus device versus a traditional device lie in the functionality, complexity, and interoperability testing. All bus systems have a digital communication protocol. While it can be argued that the protocol used by fieldbus is superior, by virtue of its ability to provide determinism for control simultaneously with prioritized event driven messages, that is not what truly distinguishes fieldbus from other nerworks. In fieldbus, *the network is the control system.* In traditional systems a *proprietary* control application resides in a centralized computing facility, either a DCS or PLC. In fieldbus the control application is *open*, and distributed across the devices on the network.

This added control functionality increases the complexity of the field devices, and the requirement for in–depth testing. But things are simpler for the end user because there is no need for custom programming on a conformant host system to make a registered device interoperate. To assure that each field device manufacturer is sufficiently conformant, and that devices from multiple manufacturers will interoperate, the Fieldbus Foundation provides a very demanding interoperability test and registration program. While this added functionality makes the development job more challenging the result is a system which is much simpler and easier for the end-user to apply, and delivers far greater performance.

DESIGN CONSIDERATIONS

Most suppliers planning the development of a FOUNDATION fieldbus device will be adapting or modifying an existing instrument. Whether this is or is not the case, there are a number of issues and trade-offs to be considered in defining the best path to a new fieldbus product. Establishing a few key facts about any existing product technology, and the manufacturer's objectives, will usually narrow the choices to a manageable few.

The comments here will not lead to a definitive recommendation because there is no one best approach for all companies. This discussion should provide a starting point, and is intended to help manufacturers recognize the general direction that may be most appropriate for their specific situation.

There are six constraints that generally bound any field device development project:

- Product Unit Cost
- Power Consumption
- Product Functionality
- Cost of Development
- Time to Product
- Project Risk

These constraints are interacting in almost every dimension. Any attempt to improve the condition of one of the constraints, will result in a penalty with one or more of the others. There are strong trade-offs among product unit cost, time to market, product functionality, and cost of development. Whether or not the device must be bus powered will affect the design effort in terms of development time and cost, and may have an impact on functionality.

The meaning of project risk is; how critical is it that the goals set forth for the other five constraints are met? An increase in development cost and time, for example, can generally be used to buy reduced risk, and vice versa. The project plan should contain clear answers or definitions for this set of constraints. A useful technique is to obtain agreement among the decision makers as to priority and, if possible, a relative prioritizing of each criterion. This will be valuable later when certain critical decisions are to be made.

Assuming the usual case, that an existing product is to be migrated to fieldbus, there are five key design decisions associated with the development project.

1. Make your own hardware interface, or buy a commercially available circuit board.
2. Use a one or two processor architecture.
3. Write your own communication stack, or buy tested, commercially available software.
4. Write your own function block application, or buy tested, commercially available software.
5. Write your own transducer block, or buy (contract) software services.

The relationships among these decisions, and the resulting work tasks, are shown in the flow chart in Figure 3. Also shown in the chart are estimates of the required engineering time, in person-weeks (P-W), for each task. The ranges reflect differences in product requirements, but assume thorough familiarity with the specifications and field device development.

The decisions themselves are somewhat interactive. A complete plan requires making a set of decisions, working out the implications, and then repeating the process to determine if a more optimal solution is possible. We will discuss the decisions and tasks overall, and then illustrate use of the chart to plan an example project.

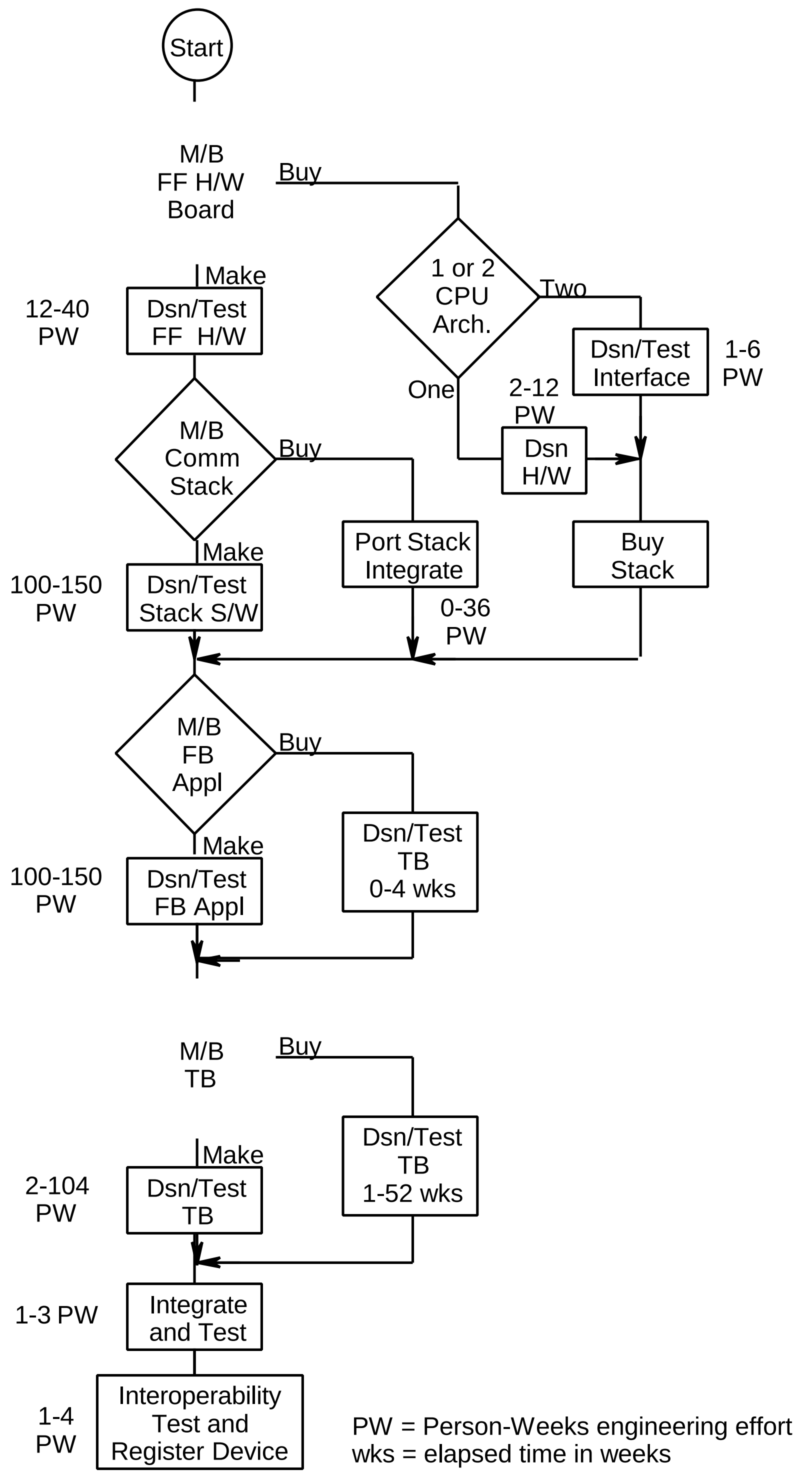

Figure 3. Flow Chart of Development Decisions and Tasks.

FOUNDATION FIELDBUS HARDWARE

The first decision to consider is whether to make or buy the circuitry that provides the interface to the bus, and provides the processing capability for both the communications and function block application.

Depending on the availability of a manufacturer's hardware design resources and the volume expectations for the product, it may be desirable to invest in hardware development to achieve minimum unit cost, minimum power consumption, and competitive or superior functionality. Alternatively, an off-the-shelf hardware solution is available which may meet all requirements. The purchase option may be the most cost effective choice if development capabilities or unit volumes are modest. Even if the analysis of all factors ultimately favors a make decision, using a purchased part may be a good interim solution because of reduced time to market, lower development cost, and lower project risk. The decision to purchase is a decision that can be changed, if and when future conditions warrant.

A decision to make obviously leads to the task of developing the required hardware. A decision to buy will lead to the next decision point.

DEVICE ARCHITECTURE

If the decision is to buy an off the shelf board, the next issue is how that board is to be integrated into the product. The base assumption is that we are migrating an existing product to fieldbus. If the existing device already uses a microprocessor for signal conditioning, diagnostics, and perhaps a communications capability such as HART™, there are two possibilities. One option is to port the existing software to the new fieldbus processor and replace some of the existing hardware. The second option is to leave the existing processor and its software relatively undisturbed, and limit the new board to supporting just the fieldbus functionality.

The first option, choice of a single processor architecture, is advantageous in terms of unit cost and perhaps power consumption. On the other hand, it may require a new board layout for the existing hardware, shown as the next task on the chart. It will also require that the existing software be moved to the new board. This work will be reflected in the "design and test transducer block" task near the end of the flow chart. The significance or cost of this work, depends entirely on the complexity of the existing functions. So, in return for a possible reduction in unit cost, the single architecture path increases the project time, cost and risk.

Selecting a two processor solution requires hardware and software interfaces between the two processors, but this is a comparatively simple task. Typically this involves connecting serial ports or sharing a dual port RAM, with appropriate software drivers. The impact on the project is the opposite of the single processor choice; this path decreases project time, cost and risk. As with each decision, the choice of a one or two processor architecture should be made in light of the agreed priorities.

Regardless of the architectural choice, the next task shown on the chart is "Buy Stack". An assumption was made here based on the following logic. The first decision made on the chart was to "Buy" the fieldbus hardware board. This was done to reduce project time and cost. It seems totally inconsistent that one would now choose to develop a communication stack for buyout hardware, when an existing tested stack has already been designed for that hardware.

COMMUNICATIONS SOFTWARE

Had the initial choice on the flow chart been to "Make" rather than "Buy" the fieldbus hardware, we would be following the path that leads to the task called design and test the Foundation fieldbus hardware (Dsn/Test FF H/W). Existing circuit designs are available for license. These can be implemented as is, or modified to suit individual requirements. A range of 12 to 40 person weeks is estimated for this task.

The next point on the flow chart is a make/buy decision regarding the communication stack. We are on this branch of the flow chart because of the initial decision to "Make" the hardware. This reflects a desire to develop a product that minimizes unit cost, and possibly to support some functionality not available on the commercial hardware. This may or may not influence the stack decision.

The skills to develop a communications stack are quite specialized and do not exist in every organization. Even if the skills do exist, stack development may not be an efficient use of such resources. If the hardware design was licensed and is being built without change, it should be possible to purchase a stack license. Without question this will be lower in cost than developing a new stack. If the hardware design was modified, then it will be necessary for the associated stack to be ported to the new hardware and re-tested for conformance. If the hardware design is extensively modified, the best choice may be to purchase source code for a stack and modify it to suit the new hardware.

If the decision is to modify or write a new stack (Make), plan on two to three person-years after the software engineers are well trained in the fieldbus specification. If the decision is to Buy, it will be necessary to either duplicate the hardware for which the stack was designed, or port the stack to the new hardware. Porting will require close cooperation from the stack supplier.

To minimize development time, cost and risk, the recommended decision here will always be to purchase a software license for an existing, tested stack. For field devices, a commercially available stack will nearly always meet requirements.

FUNCTION BLOCK APPLICATION SOFTWARE

All previous decisions in this analysis eventually lead to a make/buy decision for the function block application. The background and skills needed for developing this do not exist in every organization. Where they do exist, applying them to develop a new application is never an efficient use of resources. This is often misunderstood because the function block algorithms appear to be quite simple. But it is the interoperability requirements that present the challenge, not the control functions that a block performs in normal operation.

For example, even with a PID block, the control algorithm is less than 5% of the supporting code needed to satisfy the fieldbus requirements for mode, status, alarms, events, set point selection, and other behaviors mandatory for interoperability.

In the Make/Buy decision for the stack, hardware modifications will force porting or some other software effort on the stack. The function block application is independent of the hardware decision, assuming that the stack was properly designed to isolate the application from the hardware. So the rationale for writing a new application is even more difficult to justify than for writing a new stack.

It is not possible to develop the application for less than software can be purchased. To minimize development time, cost and risk, and to enhance performance, the recommended decision here will always be to purchase existing, tested software, even if it is necessary to port it to a new platform.

When the previously discussed decisions and tasks have been completed, the various components must be integrated and tested. When it appears that all requirements have been met, including the physical layer test[4], the final development step is to submit the device to the Fieldbus Foundation for interoperability testing. The automated test system executes over 320 test cases to determine that the device meets all interoperability requirements.

EXAMPLE ONE

We will give brief descriptions of two hypothetical situations to illustrate how the concepts presented may be used to plan a fieldbus device project. In the first example, a company manufactures and sells a complex mass flow measuring device that typically sells in the range of $7,000 to $10,000. Production quantities of the current product are about 3,000 units per year and growing. The device requires external power. It presently has a HART interface and uses HART commands for calibration and some diagnostics. Several process variables can be measured or computed with this device. The company would like to migrate this design to a fieldbus version.

Because their product is complex, and fieldbus supports such a wide range of possibilities, there is a concern that the design may get overload with unneeded features. The development strategy has therefore been determined to be; (a) implement all existing HART capabilities, but no more, (b) get a product to market as quickly as possible to gain fieldbus experience, and (c) based on the experience gained in the first year, plan a revision after 12 months.

The project constraints listed earlier will be prioritized and used as a reference in making the decisions presented in the flow chart in Figure 3. To force a spread in priorities, the highest priority will be given a rank of 10, the lowest priority will be given a rank of one, and the total ranking must equal thirty.

Since the device requires external power anyway, power consumption is elected as the least significant among these criteria. Because of the strategy, time to market is the highest priority. Since a re-design is planned for the following year, minimizing the development cost is a strong second priority. Functionality is important, but it is already defined as retaining the existing capability plus the minimum standard fieldbus capability and features.

While unit cost is always important, this is a fairly expensive device to begin with. The incremental cost of achieving fieldbus capability will be a small fraction of the total. Another consideration is that this is partly a market research initiative and the product will be redesigned in another year. Spending too much time on minimizing cost now will drive up the development cost and extend the time to market, which is contrary to the objectives.

The degree of acceptable risk is a management judgment. A value of five indicates the importance of meeting the goals defined for this project are about average, when compared to other projects this company has underway.

Based on the preceding discussion, the project planners might agree on the priority ranking shown in Table 1.

Criteria	Rank
Unit Cost	4
Power Consumption	1
Functionality	2
Development Cost	8
Time to Product	10
Project Risk	5

Table 1. Prioritized Criteria

Using the priority ranking as a reference, the decisions presented in the flow chart in Figure 3 will be resolved. Because development cost and time to market are premium requirements, and unit cost, while important, is ranked much lower, the first decision will be to buy a standard fieldbus hardware implementation. For the same reasons, a two-CPU architecture will be selected. All existing functionality will be left in the existing processor, the new fieldbus capability will be in the new processor on the

[4] Physical Layer Conformance Testing FF-830.

purchased board. The HART modem will be deleted and a serial interface will be designed with electrical isolation between the new board and the existing product.

The communication stack will be purchased, as will the function block application. The function blocks will initially include an Analog Input (AI) and a controller (PID). Others can be easily added in response to the market. Because standard hardware and software are being used, there is no porting requirement.

The transducer block is now the most significant part of the project. It will be neither a "Make" nor "Buy", but rather be a joint effort between the company's engineers and consultants who have been through the process many times before.

The magnitude of this task depends on the complexity of functions required, which are presumed to be substantial. However, the project is also bounded by the rules established regarding functionality. The transducer block software will be written use the HART protocol in communicating with the existing device. Process parameters such as mass flow, density and temperature will be passed to function blocks on the fieldbus side. Not all parameters require a function block connection. How the existing device features are mapped into the fieldbus world will be a major part of the design effort.

EXAMPLE TWO

This is an example of a company that manufactures and sells low cost on/off valves and actuators. Depending on the user's choice of solenoids, the actuators may or may not need to be bus powered. Production quantities of the existing product are in excess of 30,000 units per year. Minimum cost is critical. Functionality is initially expected to be very limited, in line with historical expectations. It is recognized that the fieldbus version of the standard product will cost more, but it is not as well recognized that it will offer added value. After some analysis, the following capabilities are identified as supporting some increase in unit cost and price. Interviews with end-users confirm that the proposed premium for the device is more than off-set by a reduction in overall system cost.

- Additional valves can be added later in the plant, without increased I/O at a central computer and without re-programming.
- Superior diagnostics are easily designed into the individual devices.
- Tag-dot-parameter locator capability simplifies commissioning compared to other bus designs.
- On/off valves are integrated with other fieldbus devices so the entire plant or operating unit uses common technology, and without feature stripping gateways.
- Simulate capability improves check out of safety and override procedures.
- Ability to run segments autonomously upon loss of host communication improves safety.
- The fault state parameter provides a standard, configurable failure mechanism within the device, in the event of a bus failure

Analysis of this company's situation yields the criteria ranking shown in Table 2. Unit cost is paramount, so development cost must be sacrificed. Low development cost and quick time to market will be sacrificed to achieve low unit cost and functional features that will lower the user's installed system cost. It is so critical that the unit cost and functionality targets be achieved, risk has been elevated to a very high rank.

Criteria	Priority
Unit Cost	10
Power Consumption	1
Functionality	6
Development Cost	2
Time to Product	3
Project Risk	8

Table 2. Prioritized Criteria

Using the rankings in Table 2, the decision on the flow chart will be evaluated as follows. The first decision will to "Make" the fieldbus hardware. As a starting point, the design for the commercially available fieldbus interface will be licensed. This is both time and cost efficient. Minimal changes will be made to the hardware and the same processor will be used in order to avoid porting standard software. This also minimizes risks that may arise when the new implementation is submitted for conformance testing. The standard communication stack for this hardware will be purchased. Likewise there are no cost or feature advantages in developing a new function block application, commercially available software will be purchased.

As with the previous example, the major software design effort is the transducer block. For an on/off valve this is inherently simple, except for the fact that the manufacturer wishes to incorporate some high value diagnostic features. The design and coding of the diagnostics will add a degree of complexity to the this phase of the project, but are critical to the strategy. Because this is their first fieldbus device, the company will use an experienced consultant to augment their own engineering team and minimize risk. The company's engineers will use their expertise in the design of unique diagnostics, the consultant will show them how to fit that functionality into a fieldbus conformant design. The consultant will also help in integrating all software, testing, writing the Device Description (DD), and in getting the product through interoperability testing.

CONCLUSION

In addition to the software license purchases, each company will require a fieldbus configuration tool, utility software for loading the stack, and a compiler. Depending on how self-sufficient a company wishes to be in the technology, there are several additional tools that may be considered.

For both examples, at the end of the project, each company will each own licenses to all standard software. They will also own source code for any custom software that was developed, the source DD, and all design documents. They consequently have control over the technology in their products.

We recommend and have demonstrated the use of priorities on the project constraints. More elaborate decision analysis techniques can be based on this approach, but is beyond the scope of this paper. Here the key decisions are illustrated in Figure 3, and the priorities of Table 1 and 2 were used to guide the decision process. Hardware development may be kept to a minimum except in situations of high volume or severe cost constraints. We recommend the use of tested, proven software whenever possible, and we have identified the major software (TB) area requiring development. It is in design of the transducer block that a company's own resources have the greatest leverage or value-add.

We recommend the use of formal, structured design methods with a written project management plan as a minimum requirement.

Finally, we advise against attempting to do the first fieldbus project without experienced technical assistance. This is not because fieldbus is inherently difficult, but it is complex, it is different, it is a *control* network. Learning how the interoperability tester interprets the specification takes time. The assistance of engineers who have been through the process will reduce that learning time several fold.