Building Plug-and-Play Networked Smart Transducers

The Proposed IEEE P1451.2 Standard Enables Transducer Manufacturers to Support Multiple Fieldbuses

1. INTRODUCTION

Several people have attempted to label the time in which we are now living as the “information age,” or the “communications age,” or even the “MTV/Wired age.” All of these “ages” share the common characteristics that they involve smart products, shared information, and communications. The label “smart” has been applied to products ranging from bombs to toasters, and has therefore lost some of its meaning. Perhaps we need a USDA definition of “smart” similar to the one proposed for “organic.”

Sensors Magazine used to use the subtitle “The Journal of Machine Perception.” I always thought that was a pretty good description of exactly what a sensor is. Perhaps you can have something called a smart product without using sensors, but it will be blind, deaf, and dumb, and will be totally dependent on others for the information that it needs to carry out its assigned tasks.

If true smart products include sensors, then what exactly is a smart sensor? While some organizations apply that label to any sensor which has a digital output, a better definition is that “Smart is as smart does.” A real smart sensor must therefore *do something* with the data, not just pass it on in a digital format. Electronics Development Corporation defines a smart sensor as a device that

- Can make decisions based on a signal input at the point of measurement,
- Can be programmed to make a variety of decisions, and
- Is able to send and receive data.

A key characteristic of a smart sensor is that it operates on the input signal in a logical fashion to increase the value of the information which it processes. The sensor is capable of making logical decisions at the source of the information. It may act on that information itself or it may pass on a high-value message, not just raw data. Other characteristics of smart sensors include their capabilities for self-test, adaptive calibration, improved rejection of spurious inputs, and ease of setup and use.

Whatever you call the age in which we are living, we are surrounded by an explosion of data, and both the technical journals and the mass media are full of discussions about the present and future need for “bandwidth.” Smart sensors help to reduce the communications bandwidth requirements by converting data into information.

The classic advantage of a smart sensor is the reduction or even elimination of the communications and control infrastructure needed to manage a system.

In the case of building or factory controls, the systems are very large and the sensor infrastructure is very expensive. Smart sensors not only provide significant savings due to the reduction in cabling and monitoring equipment costs, they provide enhanced reliability and safety because local control is maintained by the sensor even is the facility-wide control network fails.

In the case of large data acquisition systems, the needs are for reduction of cable size and weight (particularly important for aircraft and shipboard use), and for self-identification of sensors. Since the sensor channels for very large acquisition systems can number in the tens of thousands, then the reductions in the number of cable runs can be extremely important. The reduction in wiring can be a significant advantage even for smaller systems. Regardless of the size of the system, self-identification provides substantial savings in the labor for, and reduces the chance of errors during, recording the serial numbers, calibration factors, and required calibration dates for all of the sensors. The self-identification feature by itself is probably adequate justification for using smart sensors in critical data acquisition applications.

Given all of these advantages of smart sensors, why are network-compatible smart sensors not in wider use? One important reason is the somewhat fragmented nature of the existing fieldbus market and the unwillingness (or inability) of transducer manufacturers to support all of the networks now in use. Many sensor network or fieldbus implementations are currently available, each with its own strengths and weaknesses for a specific class of applications. Interfacing transducers to all these control networks and supporting the wide variety of communications protocols represents a significant and costly effort to transducer manufacturers.

There is currently no defined common digital communication interface standard between transducers and network communications nodes; each transducer manufacturer defines and builds its own interface. Consequently, transducer manufacturers cannot afford to support all of the control networks for which their products may be suited. An open, universally-accepted transducer interface standard would facilitate the development of network-capable smart sensors and actuators, and should result in faster acceptance and implementation of smart sensors into the market.

To assist in defining such a standard, the IEEE Instrumentation and Measurement Society's TC-9 Committee, in cooperation with the National Institute of Technology (NIST), conducted a series of five transducer interface workshops during 1994 and 1995 to define the objectives of such a standard. The results of these workshops was the P1451 family of proposed IEEE standards for "A Smart Transducer Interface for Sensors and Actuators." At the time of this writing, two working groups have been chartered by IEEE and have sent draft standards out for balloting by IEEE members. Two additional study groups have been formed and have submitted project authorization requests to IEEE to form formal working groups for further extensions to the P1451 family of standards.

2. IEEE P1451.2

The proposed IEEE P1451 family of standards covers network-capable smart transducers from the interface to the transducer itself up to a high-level, object-model representation of behavior, attributes, and data communications. Note that the proposed standards are written for *transducers*, a general term which includes both sensors and actuators.

The initial basic functional boundaries and interfaces defined in the proposed IEEE P1451 series of standards are illustrated in figure 1 (taken from the proposed IEEE P1451.2 standard).

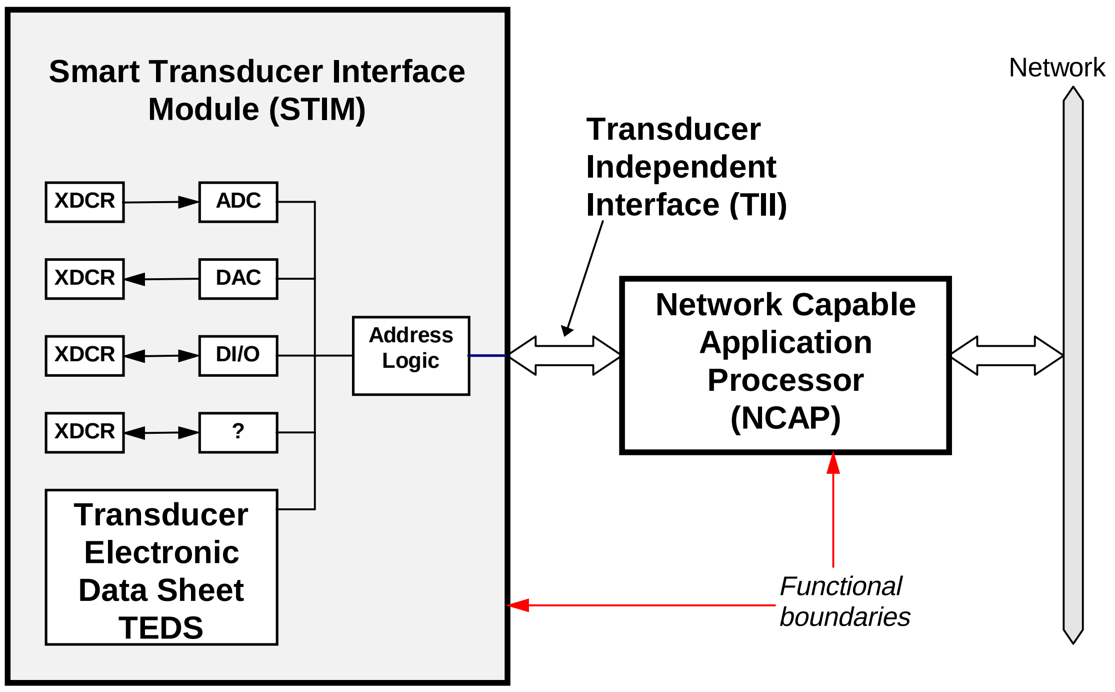
Smart Transducer Interface Module (STIM)
XDCR
ADC
XDCR
DAC
XDCR
DI/O
XDCR
?
Address Logic
Transducer Electronic Data Sheet TEDS
Transducer Independent Interface (TII)
Network Capable Application Processor (NCAP)
Network
Functional boundaries

Figure 1. Initial IEEE P1451 system block diagram

The architecture comprises a network, a Network Capable Application Processor (NCAP), and a Smart Transducer Interface Module (STIM). Note that the transducers themselves are considered part of the STIM. In fact, in order to provide the critical self-identification features, the transducer must be inseparable from the STIM electronics during normal use. For the basic system definition shown, the intended field break is the Transducer-Independent Interface (TII) between the STIM and NCAP.

The proposed IEEE P1451 standards present a high-level functional, object-oriented model of the smart transducer. This object model, which is illustrated in figure 2, consists of a series of functional blocks which can be plugged into a logical backplane to form a working system. This object model is defined by proposed IEEE P1451.1, from which figure 2 was taken.

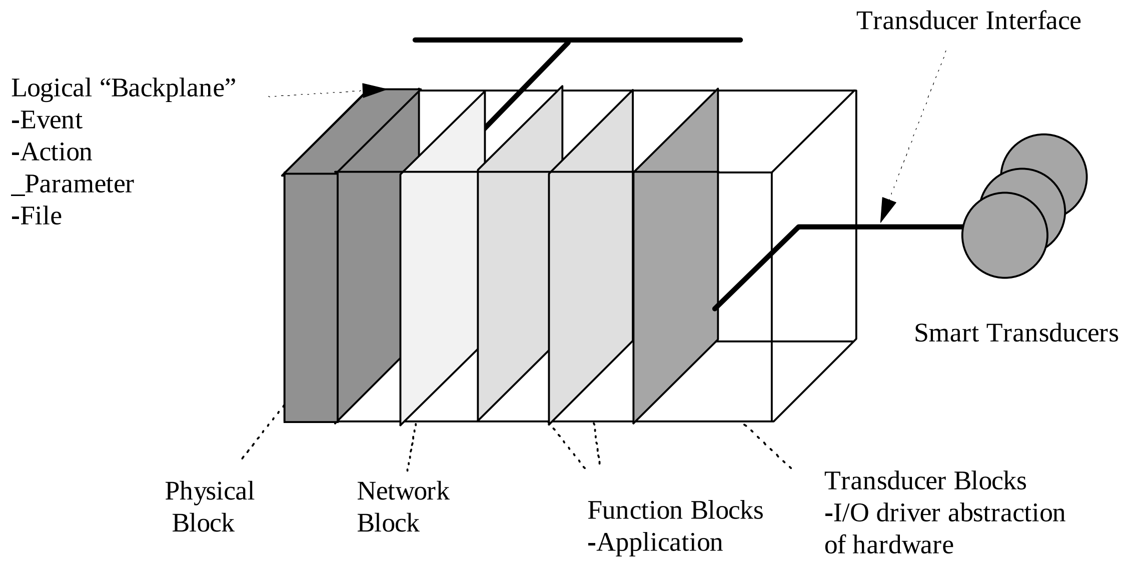

Figure 2. IEEE P1451.1 object model for networked smart transducers.

The blocks shown attached to the logical backplane make up the NCAP portion of the smart transducer. The transducer interface shown in figure 2 refers to the TII. The physical block represents the physical elements of the smart transducer, and is always there. The network block is optional, and any network can be supported within an NCAP simply by replacing the network block. Functional blocks can be included to provide special purpose functions such as Fourier transforms or PID control loops. One or more transducer blocks are present to present an abstract view of the transducers to the processor and to other nodes on the network. Each transducer block supports a TII to which a STIM can be connected.

The subtitle of Proposed IEEE P1451.1 is "Network Capable Application Processor (NCAP) Information Model." It was sent out for balloting by IEEE members in December of 1996, and the comments received from those members are being resolved. Recirculation of the draft for another ballot will be required once this is complete.

The subtitle of Proposed IEEE P1451.2 is "Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats." This draft standard covers the STIM, the TII, and the portion of the NCAP transducer block which directly communicates with the STIM. IEEE P1451.2 is further along than IEEE P1451.1. As of the time of this writing, it had completed a successful ballot of the IEEE membership, and had been submitted to the IEEE standardization committee for approval.

IEEE P1451.2 defines other critical elements of smart transducer operation and of the communications interface. These include the various formats for the Transducer Electronic Data Sheet (TEDS), the transducer functional type (sensor, actuator, buffered sensor, event sensor, etc.), and a general-purpose calibration and correction engine. The normal data format for an IEEE P151.2 smart transducer is an IEEE 754 floating-point number using standard SI units, although provision is made for using raw analog-to-digital or digital-to-analog converter counts for systems which must wring the last drop of performance out of the available hardware.

The physical connection between the NCAP and STIM is via the 10-pin Transducer Independent Interface (TII). The TII is built around synchronous serial communications based on the SPI (Serial Peripheral Interface) protocol. Several dedicated lines have been added to provide power, ground, and special-purpose control lines. The TII pin assignments are listed in table 1, which also indicates whether each physical signal is considered an input or an output by the NCAP and the STIM.

Table 1. TII Pin assignments

Pin number	Signal name	Wire color	Direction for NCAP	Direction for STIM
1	DCLK	brown	OUT	IN
2	DIN	red	OUT	IN
3	DOUT	orange	IN	OUT
4	NACK	yellow	IN	OUT
5	COMMON (GROUND)	green	POWER	POWER
6	NIOE	blue	OUT	IN
7	NINT	violet	IN	OUT
8	NTRIG	gray	OUT	IN
9	POWER (+5 VDC)	white	POWER	POWER
10	NSDET	black	IN	OUT

The TII logical signal definitions and functions are shown in table 2, which also indicates whether each signal is positive or negative logic, and whether it is level or edge sensitive. Note that the input and output naming conventions for the logical signals are from the point of view of the STIM.

Table 2. TII signal definitions.

Line	Logic	Driven by	Function
DIN	positive logic	NCAP	Address and data transport from NCAP to STIM
DOUT	positive logic	STIM	Data transport from STIM to NCAP
DCLK	positive edge	NCAP	Positive-going edge latches data on both DIN and DOUT
NIOE	active low	NCAP	Signals that the data transport is active and delimits data transport framing.
NTRIG	negative edge	NCAP	Performs triggering function

NACK	negative edge	STIM	Serves two functions: 1. trigger acknowledge 2. data transport acknowledge
NINT	negative edge	STIM	Used by the STIM to request service from the NCAP
NSDET	active low	STIM	Used by the NCAP to detect the presence of a STIM
POWER	N/A	NCAP	Nominal 5 V power supply
COMMON	N/A	NCAP	Signal common or ground

NINT is the only TII signal line that the STIM is permitted to assert at will. Other than that, the NCAP controls all communications and all message exchanges are originated by the NCAP. The other STIM outputs are not really under the full control of the STIM. NSDET is a passive signal (pin is tied to common) used to detect the presence of a STIM, and NACK is only asserted in response to an action by the NCAP. The NCAP is the SPI communications master and always controls DCLK. Even in the case of NINT, which might have been more accurately named "service request" instead of "interrupt," the resulting communications are under the control of the NCAP, which is under no timing constraint as to when it responds.

For a simple transducer, NTRIG is used by the NCAP to control the timing of when the new data is taken, in the case of a sensor, or is acted upon, in the case of an actuator. In other words, a simple sensor takes data only when triggered by the NCAP and a simple actuator updates its output only when triggered by the NCAP. More complex transducer function models (buffered sensors, data sequence sensors, event sensors, etc.) make more complex use of NTRIG. A full discussion of the defined smart transducer triggering functions is beyond the scope of this article.

From the Point of view of the NCAP, the STIM looks like a memory device. All data and commands are accessed by use of a functional address. A functional address is simply the combination of the requested service plus the channel to which it is addressed. Global communications to the STIM as a whole are addressed to channel 0. This is why a STIM is limited to 255 transducer channel instead of 256.

The basic protocol for a communication between the NCAP and STIM is that the NCAP clocks a functional address into the STIM using the DIN and DCLK signal lines. For a write, the NCAP keeps clocking DCLK and places the data on DIN. For a read, the NCAP keep clocking DCLK and looks for the data on DOUT. For all communications, NIOE serves as sort of a chip select to tell the STIM that the data transport function is active. NACK is used by the STIM to acknowledge data bytes as well as trigger signals. In order to keep the interface protocol simple, data communications and trigger functions cannot be used at the same time.

As originally written, the standard did not require a STIM to acknowledge each byte. This meant that the DCLK rate had to be slow enough so that the STIM could prepare to receive or transmit the next byte within a single DCLK period. As will be discussed, changing this to require acknowledging each byte via NACK had a dramatic effect on the achievable data transfer rate.

In addition to the logical signals briefly described above, the TII also is used for supplying power and a common (ground) reference to the STIM. All of the TII lines, including power and common, must be isolated from frame or earth ground by the STIM.

The NCAP supplies up to 75 mA at 5 V. Provision is made for supplemental power, independent of the NCAP, if needed for very sensitive or high power transducers, but power for the STIM interface control circuitry can be provided only by the NCAP. These power and ground isolation requirements are included to reduce the noise transmitted to the NCAP by the STIM, and to minimize the potential for ground loops.

One element of the definition of the 10-pin TII which appears to be missing from the above description is that of the physical connector. Then IEEE P1451.2 working group attempted to standardize on a connector, but discovered that connectors were just too application-dependent to allow forming a consensus. The applications of interest range from very cost sensitive to very high performance. This is one area where the near-universal applicability of the IEEE P1451.2 standard was almost a disadvantage.

The working group anticipates that various segments of the industry will develop *de facto* standards for the primary applications. A couple of connectors have already been used successfully. Simple 5 x 2 headers and ribbon cable have been used for demonstration systems and for benign environments such as where the TII is completely contained within a protective enclosure. Other connectors, such as 15-pin sub-miniature "D"-shell commonly used for computer video cables, are perhaps more appropriate for industrial environments and also have been used.

The IEEE P1451.2 standard is a very complex document which presents a very thorough coverage of the elements of network-capable smart transducers, and complete coverage of all of its features is beyond the scope of this article. However, two other very important elements which merit a brief description are the TEDS formats and the correction engine.

Eight different TEDS formats are defined in the standard. Of these, two are required and the other six are optional. The two required TEDS structures and two of the optional ones are defined as binary data formats and are machine-readable only. The other TEDS structures are human-readable and are stored as strings in one of several different character sets. As of the time of this writing, eighteen different languages using twenty-one different international string character sets have been enumerated in the standard, with provision for adding more as they are identified.

Defining the TEDS structures was a major part of the effort on the part of the working group. Considerable emphasis was placed on trying to identify all reasonably useful data fields to include, plus provision was made for expanding the TEDS formats by the use of industry extensions. Each TEDS structure includes a length (byte count) and a check-sum for error detection.

A brief description of the TEDS formats follows.

Meta-TEDS data block—Required, machine-readable. As defined in IEEE P1451.2, meta is a Greek prefix which means "that which pertains to the whole or overall entity, or that which is in common or shared with all member entities comprising the whole." The Meta-TEDS contains data which describe the STIM as a whole. This includes revision levels, extensions, a unique STIM identifier, worst-case timing values, number of channels, and other information regarding the STIM as a whole.

One interesting feature of the Meta-TEDS is the STIM unique identifier. The working group wanted to define a system for identifying each STIM from any manufacturer without having to resort to a coordinating committee to assign codes. The resulting identifier is based on a combination of the latitude and longitude of the manufacturing facility and the UTC code for the time of manufacture. A manufacturer may assign codes to different product lines by using slightly different coordinates for each line. The only requirement is that the coordinates must represent locations which are under the physical control of the manufacturer.

Channel TEDS data block—Required, machine-readable. The Channel TEDS defines the functional model, calibration model, physical (SI) units, upper and lower limits, timing restrictions, and any other data needed to fully describe the functioning of each transducer channel. If more than one channel are implemented, then the structure is repeated for each one.

Calibration TEDS data block—Optional, machine-readable. If the correction engine is to be used, then a Calibration TEDS must be included to define the coefficients to be used. In addition, the Calibration TEDS defines the last calibration date and time, as well as the required calibration interval, for each channel.

Meta-Identification TEDS data block—Optional, human-readable. The Meta-Identification TEDS provides the human-readable version of the identification data for the overall STIM. This information is repeated once for each language used. String fields include the manufacturer's name, the model number, the serial number, version codes, date codes, and a product description.

Channel Identification TEDS data block—Optional, human-readable. The Channel Identification TEDS includes information very similar to the Meta-Identification TEDS, except that it is for an individual channel. This is particularly useful when a STIM built by one company contains transducers which were manufactured by one or more different companies.

Calibration Identification TEDS data block—Optional, human-readable. The Calibration Identification TEDS provides a human-readable description of any information deemed relevant to the calibration of each channel. Once again, this structure is repeated for each channel and for each language supported.

End Users' Application Specific TEDS data block—Optional, human-readable. The End Users' Application Specific TEDS is provided as a place to store any additional human-readable data which is not covered by the specific TEDS fields described above. An example of an end-user field might be a description of the location of the STIM, or even the name and telephone number of the person to call for service. As for all of the human-readable TEDS, the fields are repeated for each language supported.

Generic Extension TEDS data block—Optional, readability not specified. The Generic Extension TEDS is provided to allow for future extensions to the specified TEDS. Each Generic Extension TEDS will be defined in a manner similar to that used in the IEEE P1451.2 standard, and each originating organization will have to apply for a TEDS extension ID number, to be assigned by the IEEE Standards Office. A special extension TEDS ID number is reserved for programming and testing prototypes which include experimental TEDS extensions.

The last feature of the IEEE P1451.2 standard to be described is the correction engine. Correction is the application of a specified mathematical function upon transducer data from one or more STIM channels and/or data delivered from other sources. In other words, the correction engine takes the output of one or more transducer channels, plus any other data which may be necessary, and uses a mathematical formula and one or more stored coefficients to produce a *corrected* value for the channel. The output of the correction engine is usually a floating-point number in SI units ready for communication to a user or client process.

The correction engine is reversible, in that it can also be used to convert a floating point input command for an actuator to an integer representing the required raw control input to the actuator.

The correction engine defined for IEEE P1451.2 uses a multinomial (multivariate polynomial) function with a virtually unlimited degree of the polynomial. In order to limit the degrees of the input to the correction engine, the polynomials may be segmented, that is, different sets of coefficients may be used between specified limits of the input values.

The specified correction engine is extremely powerful and has the potential to provide a standardized way of describing the calibration constants and correction coefficients for a broad range of sensors and actuators. It should be noted, however, that use of the correction engine is optional. As mentioned earlier, it is permissible for a system with limited computational capability to use a simple integer data representation rather than having to deal with floating-point mathematics.

Two other standards have been proposed for inclusion in the IEEE P1451 family and are awaiting approval from IEEE to establish formal standardization working groups. One of the proposed additional standards is IEEE P1451.3 for a high-speed multi-drop digital bus to be used for short-run connection of multiple transducers to a single STIM and NCAP. This standard will be useful for complex data acquisition systems employing large numbers of transducers distributed over a limited area and where it is not practical to run the control network or fieldbus cables to each transducer site. An example of this is the extensive instrumentation of an aircraft wing for flight testing.

The second proposed additional standard is IEEE P1451.4 for a mixed mode analog interface to be used for connecting to smart transducers through conventional analog wiring. This will be useful for adding digital TEDS capabilities to existing systems without having to replace the installed wiring. It will also be useful for very size-constrained data acquisition systems such as large arrays of very small microphones.

IEEE P1451.3 can be visualized as defining the local bus shown inside the STIM in figure 1 running between the address logic and the analog-to-digital or digital-to-analog converters. IEEE P1451.4 can be visualized as defining the analog connection to the transducer itself, except with the capability for mixing digital data with the analog signal. In both cases, the appropriate potions of the TEDS are physically located with the transducer to preserve the critical self-identification features of the overall family of standards.

3. REPRESENTATIVE STIM IMPLEMENTATION

For the Sensors Expo Boston show in May of 1997, The IEEE P1451.2 Working Group set up and operated a joint booth which demonstrated the major features and advantages of IEEE P1451.2. Hewlett-Packard Company coordinated the booth activities and provided experimental Ethernet NCAPs, the Ethernet network backbone, and the applications software and personal computers necessary to display the transducer data over the network. Hewlett-Packard Company also provided experimental STIM developers kits based on the Ethernet NCAPs to those participants who wanted to use them. These kits included experimental multi-channel analog-input STIM simulators and software support for preparing and loading TEDS structures into the STIMs over the Ethernet connection.

Additional networks besides Ethernet were planned to have been included in the demonstration booth, but the experimental Hewlett-Packard Company NCAPs were the only ones available in time for the show.

The electronics from one of the Hewlett-Packard experimental NCAPs is shown in figure 3. This small printed wiring board assembly (approximately 2” by 4”) provides the IEEE P1451.2 NCAP functions, including the TII communications interface and the floating-point correction engine, and the Ethernet network interface. For Sensors Expo Boston, the NCAP processors were programmed to produce Web pages compatible with standard World-Wide Web browser programs, including Java applications to display sensor data in trend charts.

Eleven companies plus NIST participated in the Sensors Expo Boston demonstration booth, with demonstrations ranging from simple, single-channel devices such as the one described in this article, to complex closed-loop systems using both sensors and actuators. It should be noted that, since IEEE P1451.2 was not at that time an approved standard, none of the demonstration hardware shown at Boston can be considered IEEE P1451.2 compliant. As a matter of fact, the proposed standard continued to evolve as a result of lessons learned while getting ready for the Boston show, so there are some small but important differences between that hardware and the requirements of the final draft of the proposed standard.

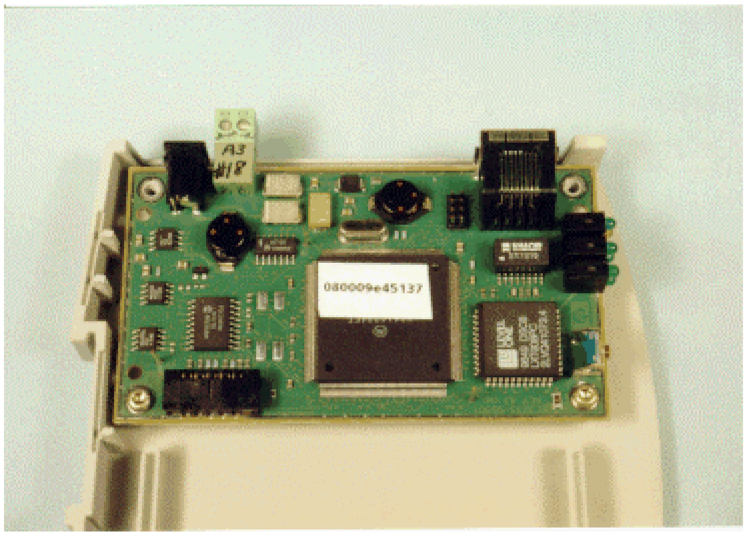

Figure 3. Demonstration Ethernet NCAP.

Electronics Development Corporation was invited in early April to build and demonstrate a miniature STIM based on our CogniSense™ EDI520 Smart Sensor Module for the show which opened on May 15. It was felt that if we could start as outsiders to the IEEE P1451.2 effort (we had not participated in the working group up until that time) and produce a working demonstration STIM in approximately three weeks, then that would serve as a powerful demonstration of the validity and maturity of the IEEE P1451.2 standard interface.

The CogniSense EDI520 Smart Sensor Module is a small (0.5 x 1.0 inch) multichip module containing a mixed-signal ASIC for the transducer interface and a microcontroller for communications and local decision-making capability. (See sidebar for description.) It was a very good choice for this demonstration project since it already contains essentially all of the hardware required to implement a STIM.

A complete STIM must include a transducer. We selected an accelerometer for this demonstration because it is easy to provide excitation. Shakers, bouncing springs, and tilt tables are all simple but visually interesting ways to stimulate an accelerometer. We used a piezoresistive accelerometer manufactured by IC Sensors because it is very small, its resistive bridge architecture directly connects to the EDI520 without requiring any additional parts, and also because we already had some in stock from another project.

The resulting miniature accelerometer STIM, which is pictured in figure 4, required approximately one square inch of printed wiring board, including the surface mounted accelerometer.

For this demonstration model, the single-channel simple sensor TEDS was programmed into the EPROM of the EDI520. By using only a portion of the optional fields in the human-readable ID TEDS, we were able to implement a complete TEDS structure in only a little over 300 bytes of data. Since these bytes were stored in the program address space of the PIC microcontroller, they were stored as 324 14-bit words in a lookup table.

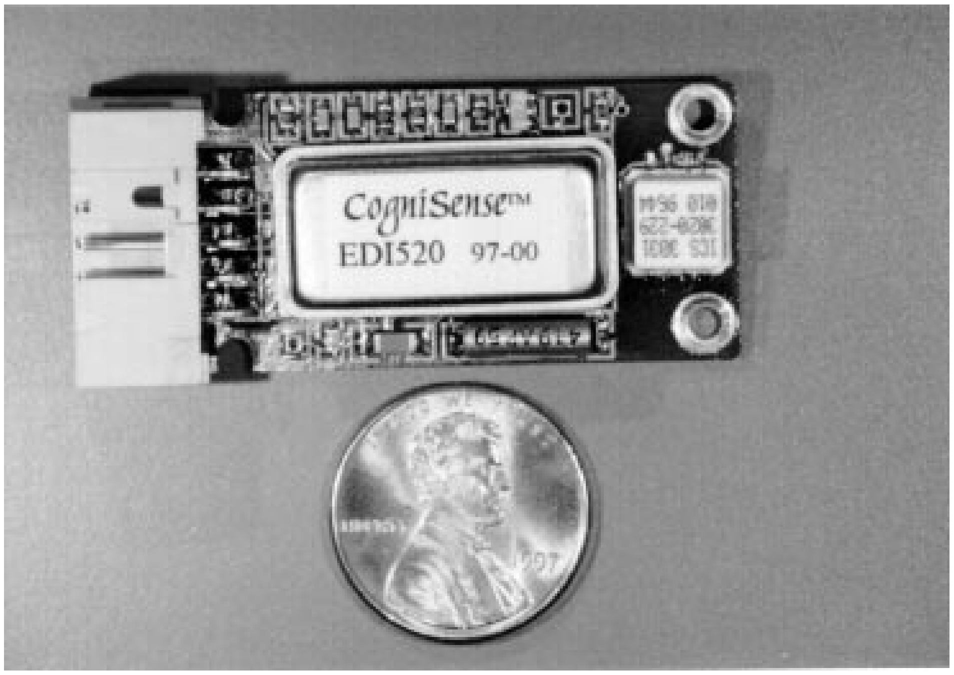

Figure 4. Miniature Accelerometer STIM

This worked quite well, especially considering that the program memory PROM is one-time programmable due to the lack of a UV-transparent window in the lid of the EDI520 module. We have previously developed techniques for changing calibration and operating constants in the PROM by writing the old data down to zeros, and entering new data in the next slot. The firmware just looks for the first non-zero entry in the lookup table and uses that. This works for a limited number of changes to the TEDS, but is not desirable for long term use where the calibration values and date will be changed at regular intervals.

Another shortcoming of using the PROM for the TEDS memory is that it can be written to only by using the in-circuit serial memory programming feature of the EDI520. This requires that a special programming voltage be applied to a certain pin that is not one of the ones used for the TII. One of the features of the proposed IEEE P1451.2 standard, which is supported by the Hewlett-Packard experimental NCAPs, is that the changeable field of the TEDS can be reprogrammed through the TII and even over the network. Using the program PROM prevents taking advantage of this feature.

A final disadvantage of using the program PROM is simply the limitations due to the physical size of the memory. The present EDI520 has 2048 words of program memory. A complex STIM with several physical and virtual data channels, and which provides the full human-readable identification TEDS fields, can use almost this much memory just for the TEDS.

Industry pundits (which always struck me as a nice way to make a living) say that we should plan for future products as if memory and bandwidth were unlimited and free. While this philosophy seems to have been embraced by the desktop computer hardware and software developers, networked industrial systems will continue to value efficiency in calculations and communications for some time. It is true, however, that a small flash or EEPROM memory for the TEDS data will enhance the utility and value of a STIM, so future versions of the miniature STIM will include at least 2048 bytes of EEPROM.

An electrical schematic of the experimental miniature STIMs built for the Sensors Expo Boston demonstration is shown in figure 5. The top portion of this schematic contains the accelerometer, the EDI520 Smart Sensor Module, and all of the components required to support those devices. The light emitting diode (LED) in the upper right corner was used to flash program status and diagnostic codes (visible on an oscilloscope, not by the naked eye) and as an acceleration-level indicator. The balance of the parts are passive components used for noise rejection and protection of the input/output ports of the EDI520.

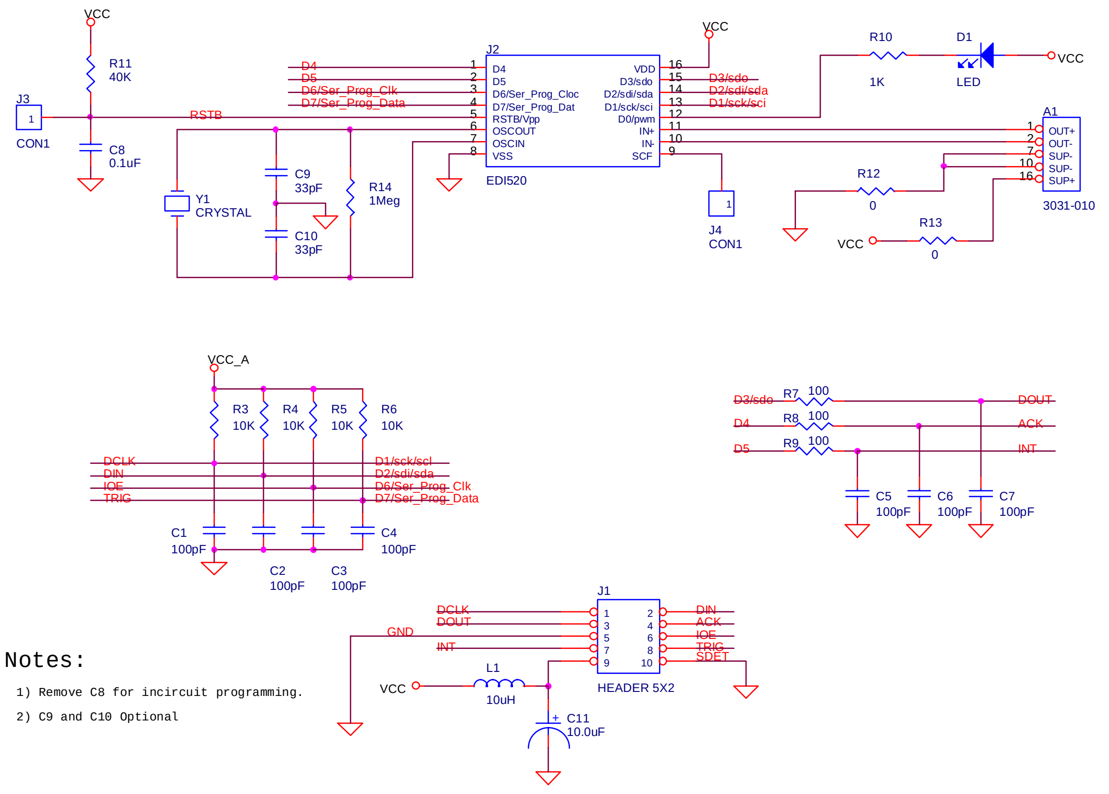

Figure 5. Electrical schematic of demonstration STIM.

Laying out the printed wiring board and assembling several of the experimental STIMs shown in figure 4 required a little over two weeks from when we started. Few problems were encountered with the physical construction and electrical portions of the TII. The only change made to the assemblies from the initial design was to increase the size of the decoupling capacitor to reduce the power supply noise generated when turning the indicator LED on and off.

Reading the acceleration level at a regular sampling interval (160 samples per second was selected as a convenient rate for the demonstration) and lighting the LED when a preset threshold was exceeded (six meters per second squared or approximately 45° of tilt) were straightforward applications of techniques developed for previous applications of the smart sensor module technology. A more interesting challenge was the firmware required to implement the proposed IEEE P1451.2 communications protocol and the TEDS.

We were pleased to discover, however, that the SPI-based TII communications protocol had been well thought out, and that the functional address scheme was regular and logical. We were able to have a prototype STIM communicating with the experimental Hewlett-Packard NCAP and transmitting a valid Meta-ID TEDS in slightly over one week after we received a copy of the draft standard and NCAP.

The EDI520 firmware for the experimental STIMs was implemented using an interrupt-driven, pseudo-preemptive real-time kernel used for previous smart sensor module projects. A complete description of the program is beyond the scope of this article, but a brief description is presented.

Independent tasks were defined for obtaining and processing data samples, for updating the gain and offset settings of the interface ASIC, for comparing the acceleration value to the preset threshold and controlling the LED, for communicating with the NCAP over the TII, and for general housekeeping. The proposed IEEE P1451.2 standard makes extensive use of state diagrams to explain the complex communications and triggering processes. Where possible, these were implemented directly in the firmware by using state variables and branch statements to produce state machine simulators.

The firmware makes extensive use of interrupts. The analog interface ASIC generates an interrupt when the next data sample is ready. The SPI port generates an interrupt when a complete byte has been received or transmitted. In addition to these event-driven task context switches, a background timer produces an interrupt for use as a "tick" to clock the real-time kernel.

One interesting design decision which had to be made was how to handle simultaneously monitoring the SPI port for a complete byte transfer, and the NIOE signal line for the start and completion of a complete message. While it may be possible to use interrupt functions for both of these signals, the primary alternatives seemed to be either using NIOE as an interrupt source to activate and deactivate a communications task which then constantly polls for complete bytes, or using the SPI byte-complete signal as an interrupt source and periodically polling NIOE for the start and stop of a message.

We chose the latter technique for the experimental STIMs because we felt that it provided a more deterministic response both to the byte-complete and data-sample-ready events. Other experimenters, including some at Hewlett-Packard Laboratories, chose the first technique because it seemed to provide better performance for their devices. The correct answer is, as always, that it depends on the application.

This selection of polling versus interrupts for the communications interface became a topic of considerable discussion due to the efforts by both Electronics Development Corporation and Hewlett-Packard to increase the data transfer rate over the TII. IEEE P1451.2 specifies a minimum data transfer rate of 6000 bits per second which all devices must support, but it permits the NCAP and STIM to communicate at any higher rate that they both agree to. Even though the hardware SPI port in the EDI520 has a theoretical capability of running at up to one million bits per second, we were unable to provide reliable communications at much higher than 6000 bits per second because of the previously-mentioned requirement that the STIM be ready with the next byte within one NCLK of the completion of the previous byte. If the previous byte were a functional address for a data read, for example, then the STIM had to decode the request and transfer the data into the transmit buffer before the next DCLK from the NCAP.

Even though there was not time to revise the communications protocol before the Boston show, this exercise did result in a concerted effort by several members of the working group, spearheaded by Hewlett-Packard, to revise the protocol to require the NCAP to hold off further DCLK transitions until the STIM acknowledged the previous byte. The NACK signal line was used for this purpose.

In informal tests of the revised protocol at Hewlett-Packard Laboratories, an Electronics Development Corporation experimental miniature STIM like the one described in this article demonstrated reliable communications with an NCAP at 500,000 bits per second with an effective data transfer rate of over 150,000 bits per second including the turnaround time between bytes.

The complete miniature STIM demonstration for Sensors Expo Boston was an accelerometer STIM mounted on a “wave machine” (a long tube filled with two heavy liquids and provided with a tilt mechanism to produce waves in the liquids). The basic arrangement is illustrated in figure 6.

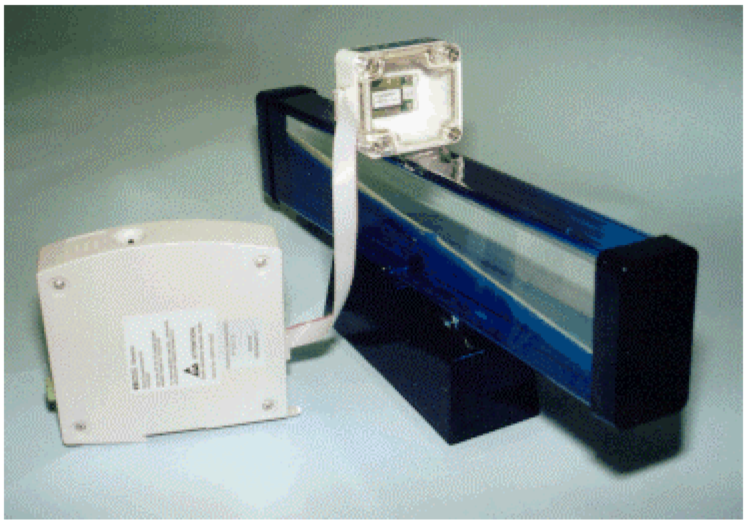

Figure 6. Sensors Expo STIM Demonstration.

The accelerometer was mounted with the sensitive axis horizontal such that it read the sine of tilt angle of the wave chamber. Accelerometers are commonly used as inclinometers in this way. The output of the accelerometer was displayed in a trend chart by the Hewlett-Packard NCAP software.

In summary of the Boston show experience, it required approximately one month total for delivery of the miniature STIMs for the demonstration, from the time we first received a copy of the proposed IEEE P1451.2 standard. This speaks extremely well of the quality and maturity of the proposed standard. We believe that it is truly ready to be issued, and that it really can be used by transducer manufacturers to produce working, network-capable but network-independent STIMs.

4. FUTURE PLANS

The proposed IEEE P1451.2 standard was submitted to the IEEE standardization committee for approval in August of 1997. It may have been approved for issue as an IEEE standard by the time you read this. Once that happens, the active members of the working group plan to aggressively pursue implementation of the standard into real systems, not just trade show demonstrations.

It is anticipated that eventually a significant number of sensors will become available with the IEEE P1451.2 interface. Transducer manufacturers will need assistance, at least at first, in applying this new, exciting, technology. To fill that need, more than one of the member companies of the working group are planning to offer assistance to transducer manufacturers for developing STIMs.

For example, Electronics Development Corporation plans to offer a complete "STIM Developers Kit" for resistive or voltage output transducers, which will include the tools to develop and manage the TEDS data, including calibration factors. Other sensor interface and microcontroller companies are expected to offer some form of assistance to transducer manufacturers who want to develop their own transducer interface circuitry and incorporate it into a STIM.

5. SUMMARY

In conclusion, the IEEE P1451.2 standard appears to be here at last. In fact, it may have been approved by the time of this publication. IEEE P1451.2 will provide transducer manufacturers the ability to produce network-capable, but network-independent smart transducers without having to become experts in each network and without having to decide how many of each network-dependent version of their products to manufacture and stock. With IEEE P1451.2, all transducers are identical regardless of the target control network or fieldbus.

An additional but sometimes overlooked benefit is that it will allow system integrators to upgrade the control network, without having to change the actual sensors. With IEEE P1451.2, not only the transducer manufacturers but also the system integrators will have more freedom from the details of the exact fieldbus implementations.

The proposed standard has been implemented by more than one company and inter-company hardware and communications compatibility has been demonstrated. By the time of this publication, we anticipate that commercial hardware will be available that is directed toward specific real-world applications.

Sidebar

A General-purpose Interface Module for Smart Transducers

The CogniSense EDI520 Smart Sensor Module used to build the demonstration STIMs described in the accompanying article will interface with a variety of resistive bridge and voltage output transducers and provides signal conditioning and decision making all in a 16-pin DIP package. The EDI520 is a multichip module consisting of a microcontroller and a mixed signal application specific integrated circuit (ASIC). The construction of the EDI520 is illustrated in figure 7.

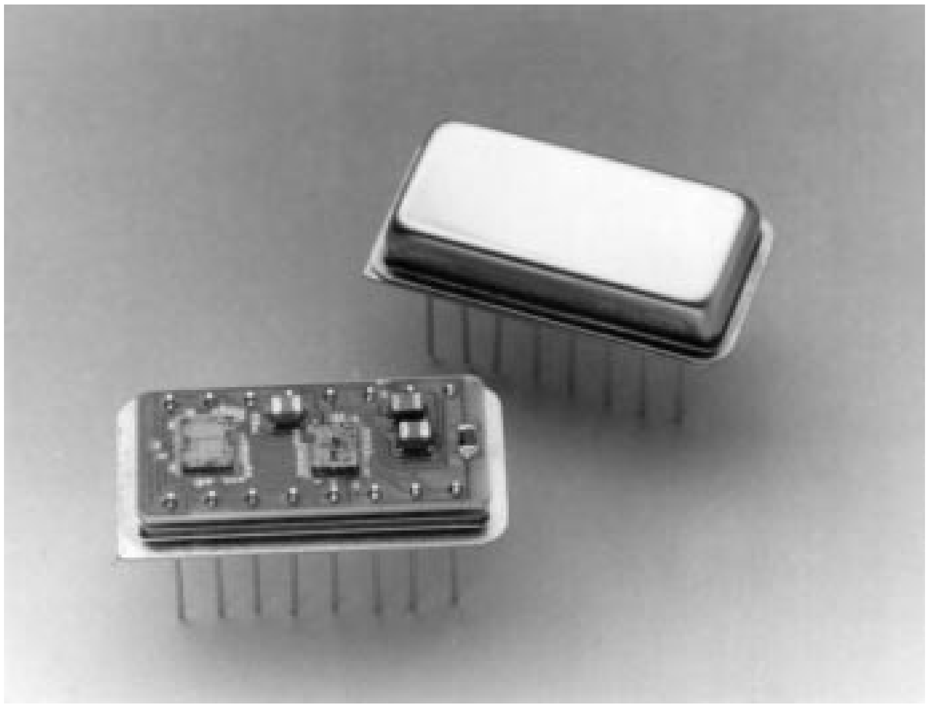

Figure 7. CogniSense EDI520 Smart Sensor Module

The signal conditioning IC comprises an instrumentation amplifier, reference adjustment digital-to-analog converter, programmable gain and frequency response filter, anti-aliasing filter, temperature sensor, 10-bit analog-to-digital converter (ADC), and a digital interface. The ADC can be multiplexed to the transducer or temperature sensor. All of the features of the ASIC can be dynamically reconfigured at any time by the microcontroller.

The microcontroller is a Microchip PIC16C64 with the I^2C/SPI port pins and the in-circuit programming pins connected to the EDI520 digital I/O. This allows the EDI520 to be custom configured for a wide range of applications and transducers. The microcontroller can be used to perform signal processing on the transducer signal such as digital filtering, threshold sensing, integration, and more.

The primary function of the microcontroller and ASIC is the compensation of the transducer for device to device and temperature induced errors. The EDI520 then allows very precise calibration of the transducer, signal processing capabilities, decision making capability and serial interface capability. The level of calibration, processing and decision making can be determined by the microcontroller software. The serial interfaces readily available with the microcontroller are the I^2C and SPI. Asynchronous communications can be implemented in software. The eight digital I/O pins also can be software programmed for any user desired function and can interface with I^2C and SPI based EEPROM memory devices for nonvolatile storage of the ASIC settings and other data sheet information.

Other versions of the CogniSense Smart Sensor Module now under development use other package styles such as the PLCC-44 for surface mounting, and include the EEPROM memory device inside the multichip module package.

Robert N. Johnson
President
Electronics Development Corporation
9055F Guilford Road
Columbia, MD 21046
410-312-6651
410-312-6653 (fax)
robertj@elecdev.com
www.elecdev.com
www.smartsensor.com