

The IEEE 1451.4 Standard for Smart Transducers

John Mark, Kistler Instrument Corporation
and
Paul Hufnagel, Kistler Instrument Corporation
Secretary, IEEE 1451.4 Standard Working Group

1. Historical Notes on Smart Sensors

Several data networks are in general use throughout industry, along with a myriad of industrial transducers, both sensors and actuators. Often, a user must turn to a single manufacturer for networks, transducers and controls, to insure compatible operation. It would make tremendous economic sense to enable any transducer to communicate using any data network. The concept of smart sensors (sensors) strives to make this a reality.

Thus, luminaries of the transducer, measurement-and-control and data-network industries discussed the concept of smart transducers, during the September 1993 IEEE TC-9 meeting (IEEE Instrumentation and Measurement Society Technical Committee on Sensors). A series of five IEEE/NIST workshops and four IEEE Working Group meetings held during 1994-1995 established the groundwork, and three committees were organized to study network connections, transducer-to-processor connections and market requirements, respectively. Two draft standards were completed for review at the end of 1995. IEEE P1451.1 Standard for Smart Transducer Interfaces dealt with the transducer electronic data sheet [1], while P1451.2 treated the common object model. [2] (*Note: This is not a typo, the groups actually exchanged numbers in early 1996.*)

The fledgling working group defined these objectives at the fall, 1995 workshop:

- a. Common communication interface between transducers and processors.
- b. Compatibility with multiple sensor actuator bus standards.
- c. Interconnect analog transducers with digital networks.
- d. Don't develop a new network standard.

The expected advantages were:

- a. Allow interoperation of transducers and control systems of different manufacturers.
- b. Allow the use of existing control system installations.
- c. Allow transducers to share a common bus, reducing wiring cost and complexity.
- d. Increase the usage of existing networks. [3]

2. Development of the smart transducer family of consensus standards

During early 1996, IEEE sanctioned two working groups to produce standards for smart transducers. IEEE P1451.1 would define a common object model description for transducers and a network capable application processor (NCAP) for network interfacing, while IEEE P1451.2 would define a transducer electronic data sheet (TEDS), smart transducer interface module (STIM) and transducer independent interface (TII).

Working group meetings held during 1996 demonstrated the need for another standard adapted to users of very small sensors and users of distributed arrays of sensors. The emerging P1451.2 STIM definition was not sufficiently flexible to allow sensors to be separated into a distributed multi-drop network, the TII was too slow to support wideband analog signals and the TEDS definition too rigid, requiring a large memory, precluding use in tiny sensors.

IEEE P1451.2 went on to become a published IEEE standard during 1997 and is formally known as *IEEE Std 1451.2-1997 IEEE Standard for a Smart Transducer Interface for Sensors and*

Actuators- Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. This standard is presently (ca. 2004) in the process of revision, to allow wider acceptance and usage.

IEEE 1451.1 was completed and published during 1999 as *IEEE Std 1451.1-1999 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators- Network Capable Application Processor Information Model*.

Following study group meetings through late 1996 and 1997, the IEEE sanctioned two new working groups, one for multi-drop sensors (IEEE P1451.3) and another for mixed-mode analog sensors with compact TEDS (IEEE P1451.4), late in the fall of 1997.

IEEE 1451.3 evolved to define a multi-drop sensor data network based on an RF-spread-spectrum-in-wire physical medium and was published in 2003 as *IEEE Std 1451.3-2003 IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems*.

IEEE P1451.4 was accepted as a full-use standard by the IEEE Standards Association, May 14, 2004, and is available as *(D3.0) Draft Standard for A Smart Transducer Interface for Sensors and Actuators - Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, until final publication, planned for the second half of 2004.

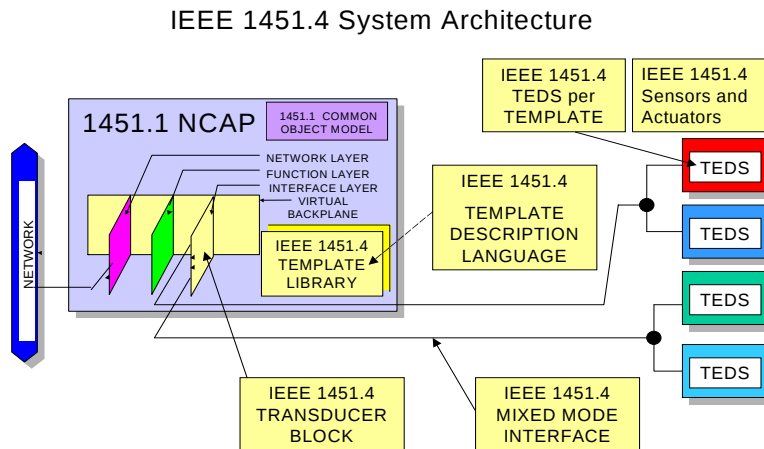
3. IEEE 1451.4- In General

IEEE 1451.4 defines a physical connection (Mixed-Mode Interface, or MMI) that is alternately used for TEDS data and analog signals, on either 2, 3 or 4 wires. This adapts the Standard for use with a wide variety of sensors and actuators.

Contained in the standard is a definition for a transducer electronic data sheet (TEDS), which allows the use of very small memories, through the use of templates. The small physical size of the memory device allows TEDS to be included in tiny, lightweight sensors. However, the low bit-count available in physically small memory devices dictates that only essential data be stored in an array governed by a template. The template defines the significance and units associated with the stored data and the mapping of the data in memory. Prior to storing data and upon reading back the stored TEDS data, the template guides, respectively, packing and unpacking the data. New templates may be written in the template description language (TDL) defined by the standard. [4]

The diagram below describes the relationship between the parts of IEEE Standard 1451.4-2004.

Figure 1



4. IEEE 1451.4- Usage

Due to its use of templates, IEEE 1451.4 allows TEDS data to be stored in memories that are physically small, extending the use of the standard to small lightweight transducers of many descriptions. Templates may be written for transducers not yet defined in the template library contained in the standard, by using the TDL. This type of development is encouraged.

The MMI is defined in two classes, again to allow as broad as possible a selection of transducers to utilize the Standard. Class 1 of the MMI definition, to minimize system wiring, defines the use of a single connection by both analog signals and data, alternately. Class 1 is well adapted to accelerometers and microphones containing current operated amplifiers. Meanwhile Class 2 allows for analog and data to be transferred via separate connections, in applications not adaptable to a shared connection. Class 2 applications might include very high impedance sensors (pH glass electrodes), actuators and sensors of the 4-20mA variety (I/P control pressure regulators, for example, popularly known as Ida Pease) and bridge transducers, thermistors or RTD temperature sensors, which would have errors added by switching diodes.

In general, the aim of the IEEE 1451.4 Standard is adaptability to allow TEDS to be used in every form of transducer.

In general, data contained in the TEDS includes transducer type and manufacturer identification (Basic TEDS), to allow a data system to choose the correct template for the interpretation of the stored data. The body of the TEDS data includes calibration information, which allows a "smart" data system to adjust for variations in transducer sensitivity, standardizing its response. A small user area allows transducers in a large array to be identified by location, potentially easing the burdensome bookkeeping in such areas as experimental modal analysis.

Finally, the Standard defines a transducer block (T-block), which contains the "hooks" needed to connect a 1451.4 system into a 1451.1 network capable application processor.

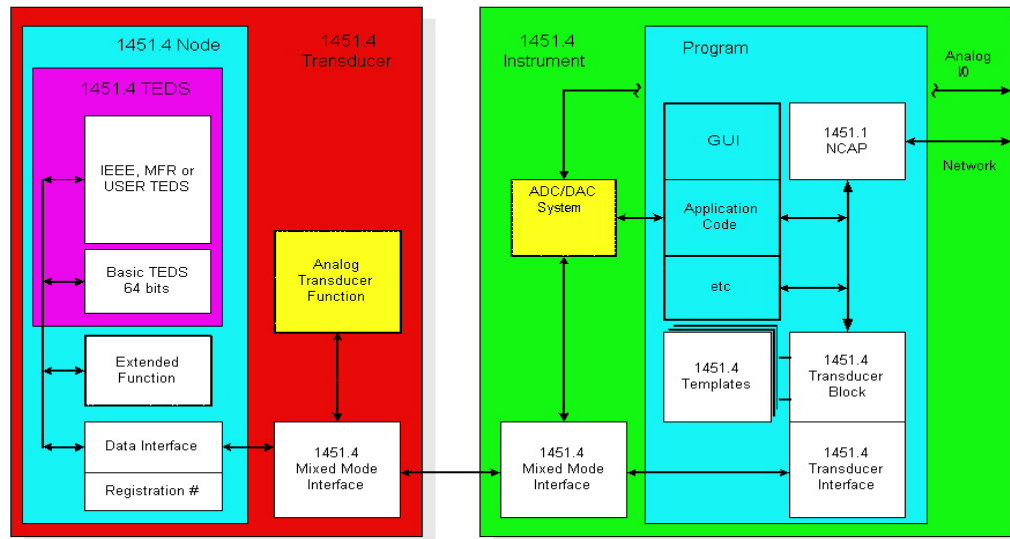
5. IEEE 1451.4- System Details

The IEEE 1451.4 TEDS System Flowchart below describes a hierarchy of functions necessary to house the TEDS data within a physical memory, and to allow accessibility of that data, via a data transmission protocol. The flowchart illustrates all parts of a fully implemented IEEE 1451.4 compliant TEDS system, in which the TEDS is contained within a node and the node located

inside a transducer, which contains a Mixed Mode Interface. TEDS data are available to a measurement/control device containing the 1451.4 Transducer Interface and Transducer Block and a Template Library. These three functions are firmware objects, which fit into the IEEE Standard 1451.1 NCAP Object Model, described fully in *IEEE Std.1451.1-1999*.

Figure 2

IEEE 1451.4 TEDS System Flowchart



Subsets of this implementation are recognized, under IEEE 1451.4, including a minimal implementation, in which the IEEE 1451.4 TEDS has been assimilated into the Transducer Block, along with a single Template. This implementation might be the case for a small instrument containing an integral transducer.

Included, for reference, in the block diagram are functions not defined in this standard, but which may be found in a typical application. Following are brief descriptions of these functions:

Within the IEEE 1451.4 Transducer, an analog transducer shares the Mixed Mode Interface.

TEDS data enters or exits a 1451.4 Node via the data interface, which is a control and transmit/receive function. Each transaction through the data interface is qualified by the unique registration number (URN), which controls communication with individual nodes connected to the MMI. The URN contains a family code, a serial number and a CRC code. The family code primarily defines the command set used when communicating with the device. The serial number and CRC allow unique access to an individual node device connected in a multi-drop system. [5]

Along with TEDS memory, a node may contain additional functions, such as switches, counters, temperature measurement, etc. as represented by the extended function block. Access for these functions will be found in the device specification, as implied by the family code.

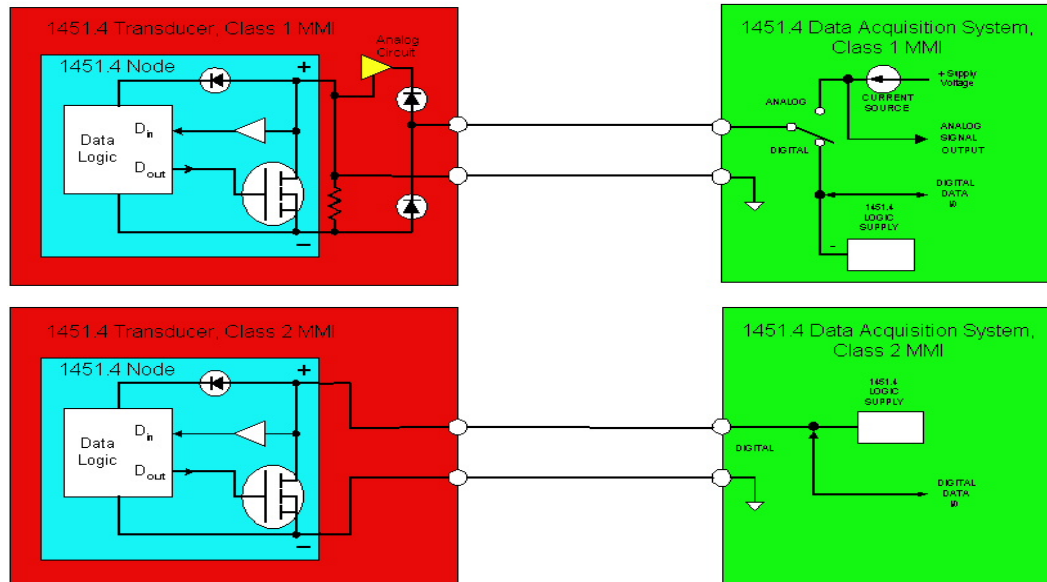
Within the 1451.4 instrument, analog data enters or leaves the application program block through an analog system, which shares the 1451.4 MMI. All IEEE 1451.4 functions are parts of the overall program code, which may also include application program code for the specific function of the instrument, a graphic user interface (GUI) and display, network interface, and so on. The 1451.4 Transducer Interface contains a low-level command structure necessary to execute the communication protocol required by the TEDS nodes.

6. IEEE 1451.4- Mixed Mode Interfaces, Class 1 and 2

The IEEE 1451.4 Standard defines two classes of MMI, to allow the standard to be used with as broad a selection of transducers as possible. Details of the circuit operation for IEEE 1451.2 Class 1 and Class 2 TEDS devices are shown in the diagram below. Although sensors are shown below, actuators are equally able to communicate using the 1451.4 MMI.

Figure 3

IEEE 1451.4 Class 1 and 2 MMI Details



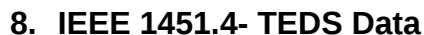
Class 1 defines sequential sharing of a single connection, comprised of two wires, by analog and digital signals. An analog signal is defined as a positive voltage with respect to signal return, while digital data is transmitted as a negative voltage. For Class 1, zero volts is a logic zero and -5 volts is a logic one.

Class 2 defines a data connection independent of the analog signal, allowing TEDS to be included in applications which are not adaptable to shared operation between data and analog signals, such as bridges, thermocouples, current operated transducers, etc., which cannot have a switching diode in the analog signal connection. For Class 2, zero volts is a logic zero and +5 volts is a logic one. Because the connections are independent, analog signals and digital data are available simultaneously and continuously, in Class 2.

7. IEEE 1451.4- Mixed Mode Interface Data Transfer

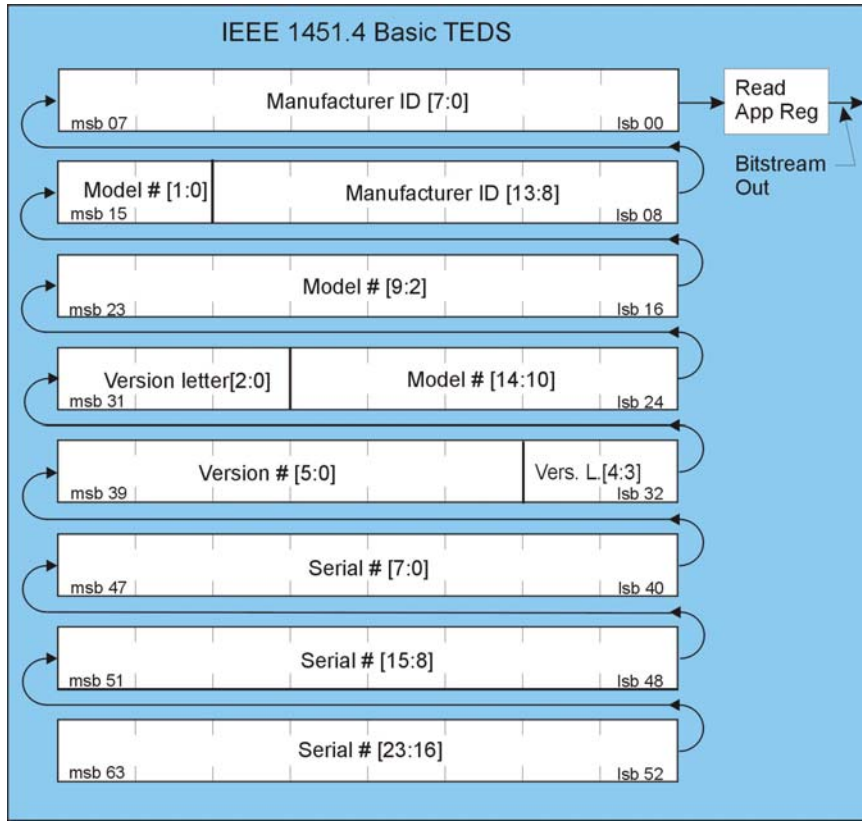
Data transfer in the mixed mode interface is implemented via commands made up of four pulse intervals: reset, write-one, write-zero and read. These are applied in a sequence defined for the TEDS node device. Transfers always begin with a reset interval, followed by a ROM command, made up of read and write sequences, followed by a RAM command, also comprised of read and write sequences. The sequences for Class 1 are shown in the timing diagrams below, with pulses between zero and negative 5 volts, nominally. The signal polarity is reversed for Class 2, with pulses between zero and positive 5 volts. Details of the command structure may be found in the voluminous Maxim/Dallas Semiconductor literature on the *Dallas Semiconductor iButton Protocol*. [6]

IEEE 1451.4 MMI Data Transmission Waveforms



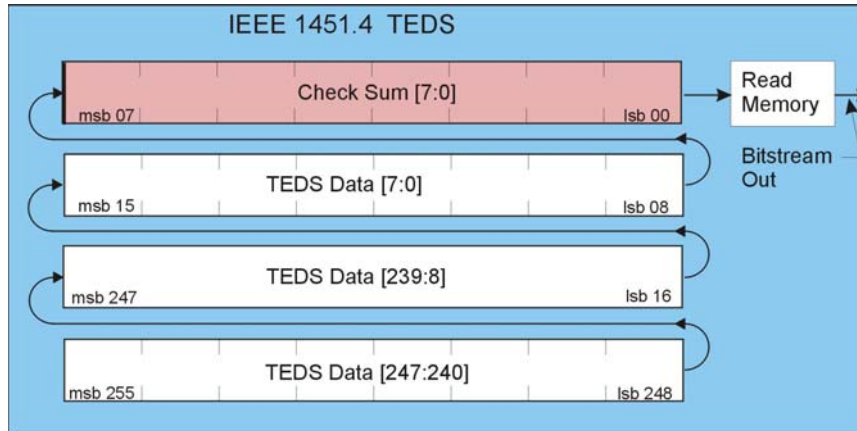
Many IEEE 1451.4 transducers will incorporate a simple DS2430A, or equivalent, memory device. This device contains a 64 bit write once memory called the application register, an eight bit status register, and a 256 bit EPROM called the data memory. The eight-bit status register is used to determine whether or not the application register has been burned. [7] According to the standard, the 64 bit Basic TEDS must be stored in the application register. The rest of the data is stored in the data memory. [8]

Figure 5 Basic TEDS bit map



Checksums are used to ensure that data is transferred correctly over the digital bus. In general, an eight-bit checksum is computed for each 256-bit section of memory in a node. For the DS2430A, since there is only one 256-bit memory, there is only one checksum, which is located in the first byte of the data memory. This is labeled byte 0 in the description of the data memory for the DS2430A. However, since the application register also contains data and can only be written once, the checksum in the data memory also includes the Basic TEDS stored in the application register. The checksum is calculated by first summing each byte in the memory and taking the two's complement of the lowest eight bits of the sum. This way, an eight-bit sum of all the data and the checksum should equal 0. After the checksums have been verified, they are discarded.

Figure 6 IEEE TEDS bit map



The data in the memory device is packed into a true bit stream. There are no word length boundaries that are established by the structure of the data itself except for the structure of the basic TEDS, which was chosen to fit within the 64-bit width of the application register. All of the fields within the basic TEDS and any templates that follow are only word aligned if the designer of the template chooses to do so. Each byte that is read from the memory device is packed into the bit stream starting with the least significant bit and ending with the most significant bit of the byte. As bits are read from the stream, the first bit out is the least significant bit of the current value being read. If a TEDS data field is six bits wide, six bits are removed from the bit stream and the first bit is placed in the least significant position while the last bit removed is placed in the most significant position.

In the case of a transducer containing a single DS2430A, assembling the bit stream is performed as follows. The transducer block begins by reading the bytes from the application register starting from byte 0. The data memory is concatenated in the same way after the checksum byte is discarded. In this case, that means that byte 1 of the data memory follows byte eight of the application register. The last byte in the stream is byte 31 of the data memory. In all, with the checksum discarded, there are 39 bytes in the stream.

At this point, the basic TEDS can be examined. The first 14 bits are removed from the bit stream as described above. These bits represent the manufacturer of the transducer. The rest of the basic TEDS is removed as shown in the table below. The value of the manufacturer ID is described in Annex J of the standard. The version letter is a Chr5 type that is described in section 7.4.5.2.5. [9]

Basic TEDS		
Field	Size	Range
Manufacturer ID	14 bits	17-16381
Model Number	15 bits	0-32767
Version Letter	5 bits	A-Z (Chr5)
Version Number	6 bits	0-63
Serial Number	24 bits	0-16777215

The rest of the TEDS data is divided into major sections. These sections are typically made up of templates. At the beginning of each section is a two-bit field called the "Selector of Descriptor" or DS. This field is used at the beginning of every major block of data and is described in more detail in Section 6.4 of the standard. [10] For a block that is described by an IEEE Template, the value of the field is always zero and the next eight bits are used to determine which template describes that block of data. A TEDS data stream is often ended with a 7-bit ASCII data block; however, any number of templates may follow provided there is enough room in memory to

accommodate them. A block of 7-bit ASCII data is indicated by a DS set to 3 followed by one bit set to 1. The data in this kind of block is meant to be simple text data that can be easily edited by the user.

9. IEEE 1451.4- Templates

IEEE 1451.4 templates are stored in text files. Each file may contain any number of templates and the entire file is secured with a checksum so that its authenticity can be verified. Templates are written using the template description language (TDL) described in Section 7.0 of the standard and are used to encode and decode the data stored in the TEDS memory. The IEEE 1451.4 standard templates are included in Annex A of the standard. [11]

Essentially, an IEEE 1451.4 template is a set of properties or values that describe key characteristics of the transducer. For example, many sensors are characterized by a property called the sensitivity. The sensitivity is often a simple value used to convert the voltage seen at the output of the sensor into the physical units being measured and is often expressed as a ratio of volts to physical units. The TDL provides several ways to encode values into the TEDS memory with varying degrees of precision and compression. Along with unsigned integers of any length, the TDL supports four byte IEEE floating point values and two integer encoded floating point values that provide constant resolution and constant relative resolution. All of the data types are described in Section 7.4.5. [12] A template could specify that a sensitivity for an accelerometer is stored using the units V/m/s/s in the TEDS memory as a four byte IEEE floating point value.

Statements in a template usually refer to data that is stored in a TEDS memory; however, they do not necessarily need to. All statements that do interact with the memory specify the number of bits that are used. As an example, the sensitivity of a sensor will be directly related to a particular sensor and therefore the information will be stored in the TEDS memory using a specified number of bits. On the other hand, a property that describes the type of electrical output signal could be the same for all sensors using a specific template and the value of the property could be specified in the template itself. This type of property would not require any bits to be read from or written to the TEDS memory.

A TEDS template also allows the designer the ability to suggest three different levels of access to information within the TEDS memory. The access level ID, Cal or Usr is specified as a parameter on the statement if a keyword supports an access level. The ID level is primarily used for manufacturers of transducers. This information is considered to be integral to the transducer and includes all of the information in the basic TEDS. The Cal level specifies that the information should be changed only when the sensors calibration information has changed. The Usr is used for information that is user defined and can be changed at any time.

As an introduction to the language, the IEEE 1451.4 Standard Template 25 will be used as an example. [13] The template begins with the TEMPLATE keyword as such:

```
TEMPLATE 0,8,25,"Accelerometer and Force Transducer"
//The first 0 in the Template field indicates IEEE ...
//the sensor to get the template ID, the 25 is ...
TDL_VERSION_NUMBER 2
ABSTRACT IEEE 1451.4 Default Accelerometer and Force Transducer Template
SPACING
```

The TEMPLATE keyword requires four parameters. The first parameter indicates the publisher of the template and must be a valid manufacturer ID; in this case the 0 indicates the template is a standard template. The next parameter indicates how many bits are used to hold the template ID that is specified in the third parameter. The eight bits are read from the TEDS data and compared to the template ID of the template. If the value is 25, then this template is used to decode the rest of the data. If not, then another template must be found. All of the standard

templates use eight bits to signify the template ID. While designing a template care must be taken to make sure that a template ID can be uniquely determined. It is often a good idea to use eight bits to hold a template ID. The last parameter is a text string that is used to name the particular template.

The next two lines are comment lines intended only to describe some feature of the template to anyone who may be reading through the text of the template. Comments begin with the double slash and continue to the end of the current line. Many comments in the current template have been removed for clarity during this discussion.

Statement	Description
TDL_VERSION_NUMBER	Indicate the version of the TDL that was used to publish the template. All templates published to the standard (TDL v1.0) will use the value 2.
ABSTRACT	All text following an ABSTRACT keyword is concatenated and used to annotate the template. The contents of the abstract are purely informational and are intended primarily for use in software that is designed to work with templates in a functional capacity.
SPACING	Intended as a hint to software that is displaying the contents of a TEDS memory to provide some sort of separation between the property above and below the current line.

At this point, only the TEDS template ID has been read from the TEDS data. None of the other keywords actually causes data to be read.

```
//Physical Base Units: (ratio, radian, steradian, meter, kg, sec, Amp, kelvin,
mole, candela, scaling, offset)
PHYSICAL_UNIT "V/(m/s2)", (0,0,0,1,1,-1,-1,0,0,0,1,0)
```

Properties often require a set of mechanical units in their description. The PHYSICAL_UNIT keyword allows the use of any unit that can be expressed in terms SI units. The statement associates a string that describes the unit to an encoded form of the unit that is more easily converted among other units of the same class, such as pounds to newtons and meters to feet. The string representation, once it is declared can be used on properties described in the template. The use of physical units is described in more detail in Section 7.4.7. [14]

```
SELECTCASE "Transducer Type", ID, 1
CASE "Accelerometer", 0
...
statements removed
...
ENDCASE
CASE "Force Transducer", 1
...
statements removed
...
ENDCASE
ENDSELECT
```

The SELECTCASE keyword begins a block of the template that will be dependent on a value stored in the TEDS memory. Within the SELECTCASE/ENDSELECT block, there will be a number of CASE/ENDCASE blocks. Only one CASE/ENDCASE block will be evaluated based on the value in the TEDS memory. The width of the field determines how many cases may be inserted. In this example, one bit is allocated which will allow two distinct cases. If the TEDS memory is being decoded, the single bit is read from the memory and if it is a 0, the case named “Accelerometer” is activated; while if it is a 1, the “Force Transducer” case is used.

The differences between the Accelerometer case and the Force Transducer case are minimal and only the accelerometer will be examined. The next SELECTCASE determines whether or not the transducer supports “Extended Functionality”. Along with devices that contain memory, the standard also allows devices to be inserted into a transducer that can change the behavior of the

transducer. Some devices are already in use that provide electrical switches which can control the sensitivity of a sensor. The TDL provides a means to describe situations where these devices may be used. This aspect of the TDL is beyond the scope of this discussion and that part of the template will be skipped over. More information can be found in Section 7.4.9 of the standard. [15]

```
SELECTCASE "Extended Functionality", ID, 1
CASE "None", 0

UGID "I25-0-0-0", "Accelerometer"

%Sens@Ref, "Sensitivity @ reference condition", CAL, 16, ConRelRes, 5E-7,
0.00015, "rp", "V/(m/s²)"
%TF_HP_S, "High pass cut-off frequency (F hp)", CAL, 8, ConRelRes, 0.005, 0.03,
"rp", "Hz"
ENDCASE

CASE "Programmable sensitivity", 1
...
ENDCASE
ENDSELECT
```

The UGID keyword is used to group similar templates. This identifies templates that apply to the same type of device and therefore may be interchanged.

Properties are denoted in the TDL by the character “%”. The first property that has been encountered in this template is the “Sens@Ref” property. All reserved properties are listed in Annex B.1. Manufacturer defined properties should begin with the prefix “%MDEF”. The Sens@Ref property in this example is store in 16 bits and uses the Constant Relative Resolution encoding (see Section 7.4.5.3.4). The last two parameters of the statement are a format specifier (Section 7.4.6) and the physical units.

```
ENUMERATE DirectionEnum, "x", "y", "z"
%Direction, "Sensitivity direction (x,y,z)", CAL, 2, DirectionEnum, "e", ""
```

The Direction property illustrates an example of an enumerated property. The ENUMERATE statement defines a set of values that are assigned values beginning with 0 and increasing by one for each element. In this case, the assignments are “x” = 0, “y” = 1, and “z” = 2. Any property using this enumeration must allocate at least 2 bits of storage to accommodate all possibilities.

```
%Weight, "Transducer weight", CAL, 6, CONRELRES, 0.1, 0.1, "rp", "g"

ENUMERATE ElecSigTypeEnum, "Voltage Sensor", "Current Sensor", "Resistance
Sensor", "Bridge Sensor", "LVDT
Sensor", "Potentiometric Voltage Divider Sensor", "Pulse Sensor", "Voltage
Actuator", "Current Actuator", "Pulse Actuator"
```

```
%ElecSigType, "Transducer Electrical Signal Type", ID, 0,
ElecSigTypeEnum, "e", "" = "Voltage Sensor"
```

The ElecSigType property shows an example of a property that is defined strictly by the template. The width field of the property is set to 0 to indicate that no bits are to be read from the TEDS memory and the property is followed by an assignment. The ElecSigType property of this sensor will always evaluate to “Voltage Sensor”.

```
ENUMERATE MapMethEnum, "Linear", "Inverse m/(x+b)", "Inverse (b+m/x)", "Inverse
1/(b+m/x)", "Thermocouple", "Thermistor", "RTD", "Bridge"
%MapMeth, "Mapping Method", ID, 0, MapMethEnum, "e", "" = "Linear"
ENUMERATE ACDCCouplingEnum, "DC", "AC"
%ACDCCoupling, "AC or DC Coupling", ID, 0, ACDCCouplingEnum, "e", "" = "AC"
ENUMERATE SignEnum, "Positive", "Negative"
```

```
%Sign, "Polarity (Sign)", CAL, 1, SignEnum, "e", ""
```

```
SELECTCASE "Transfer Function", ID, 1  
CASE "Not specified", 0  
ENDCASE
```

A CASE with an empty body is perfectly legal and may be significant. If a case does not exist for a given value, the case is assumed to be empty.

```
CASE "Specified", 1  
%TF_SP, "Low pass cut-off frequency (F lp)", CAL, 7, ConRelRes, 10, 0.05, "rp",  
"Hz"  
%TF_KPr, "Resonance frequency (F res)", CAL, 9, ConRelRes, 100, 0.01, "rp",  
"Hz"  
%TF_KPq, "Quality factor @ F res (Q)", CAL, 9, ConRelRes, 0.4, 0.01, "rp", ""  
%TF_SL, "Amplitude slope (a)", CAL, 7, ConRes, -6.3, 0.1, "0.0", "%/decade"  
%TempCoef, "Temperature coefficient (b)", CAL, 6, ConRes, -0.8,  
0.025, "0.000", "%/°C"  
ENDCASE  
ENDSELECT
```

```
%Reffreq, "Reference frequency (F ref)", CAL, 8, ConRelRes, 0.35, 0.0175,  
"0p", "Hz"  
%RefTemp, "Reference temperature (T ref)", CAL, 5, ConRes, 15, 0.5, "0.0", "°C"  
%CalDate, "Calibration Date", CAL, 16, DATE, "d-mmm-yyyy", ""  
%CalInitials, "Calibration Initials", CAL, 15, CHR5, "s", ""  
%CalPeriod, "Calibration Period (Days)", CAL, 12, UNINT, "0", "days"  
%MeasID, "Measurement location ID", USR, 11, UNINT, "0", ""
```

```
ENDTEMPLATE
```

All templates must end with the ENDTEMPLATE keyword. This allows multiple templates to be combined into one text file.

-
- [1] El-Kareh, E. V., editor, "IEEE P1451.1 Standard for Smart Transducer Interface, Transducer Electronic Data Sheet and Transducer Interface Specification, Draft 1.40," IEEE Standards Association, 11-16-95.
- [2] Eidson, John C., editor, "IEEE P1451.2 Standard for Smart Transducer Interface, Object Model Specification, Draft 1.61," IEEE Standards Association, 11-02-95
- [3] Lee, Kang B., "Opening Remarks," *Fifth IEEE/NIST Workshop on Smart Transducer Interface Standards*, NIST, Gaithersburg, Maryland, November 15, 1995.
- [4] Jones, Charles H., "IEEE 1451.4 Smart Transducers Template Description Language," May 2004.
- [5] Data Sheet, "DS2430A 256-Bit 1-Wire EEPROM," Maxim/Dallas Semiconductor, edition 02-20-02.
- [6] *Book of iButton Standards*, Maxim/Dallas Semiconductor, Appendix 1, pp.125-129, edition 08-12-97.
- [7] See endnote [5], pp. 5, 7, 8.
- [8] "IEEE Std 1451.4-2004, Standard for a Smart Transducer Interface for Sensors and Actuators- Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats", IEEE Standards Association, Piscataway, NJ, subclause 5.1.2, 2004.
- [9] See reference [8], subclause 7.4.5.2.5.
- [10] See reference [8], subclause 6.4.
- [11] See reference [8], Annex A.
- [12] See reference [8], subclause 7.4.5.
- [13] See reference [8], Annex A, Table A.2.
- [14] See reference [8], subclause 7.4.7.
- [15] See reference [8], subclause 7.4.9.