

Capítulo 1

IEEE 1451

El modelo IEEE 1451 no propone simplemente un protocolo de red. En su lugar, establece un conjunto de interfaces para hardware y software con el fin de separar el diseño de los transductores de la elección de las redes de comunicación. Al separar las dos entidades -transductores y redes- el modelo permite que el fabricante se centre en su dispositivo de transducción, sin tener que preocuparse de su adaptación a diferentes redes, lo que podría contribuir a mejorar la calidad de los transductores y a reducir sus precios.

La familia IEEE 1451 define un conjunto de interfaces de comunicación para conectar transductores inteligentes a sistemas basados en microprocesadores, instrumentos y redes; y proporciona un conjunto de protocolos para sistemas tanto cableados como inalámbricos. La Figura 1.1 muestra la arquitectura completa. Básicamente, se define un *Network Capable Application Processor* (NCAP) y un *Transducer Interface Module* (TIM), además del interfaz entre ambos. El NCAP consiste en hardware y software cuya misión es proporcionar una pasarela entre los distintos TIMs y la red de usuario, además controla al TIM a través de un interfaz digital.

El TIM consta de uno o varios transductores, electrónica de conversión y procesado de señal, hojas de datos llamadas *Transducer Electronic Data Sheets* (TEDS) y un módulo de comunicación IEEE 1451.X referido al interfaz físico entre éste y el NCAP. Los TEDS contienen información tal como la identificación del transductor, su rango de medición o datos de calibración.

Para acceder al NCAP a través de la red de usuario existen tres posibles opciones:

- El protocolo IEEE 1451.1.
- El protocolo HTTP tal como aparece en la norma IEEE 1451.0.
- Una propuesta de servicios WEB, *Smart Transducer Web Services* (STWS) [1].

Entre las diferentes ventajas de IEEE 1451 se pueden citar las siguientes:

- Es lo suficientemente completo como para cubrir la gran mayoría de sensores y actuadores disponibles.
- Dispone de diversos modos de operación (agrupación de canales, diferentes modos de disparo, etc.).
- Es compatible con varios tipos de buses y redes (tanto cableadas como inalámbricas).
- Trabaja con un protocolo binario eficiente (de importancia para redes inalámbricas).

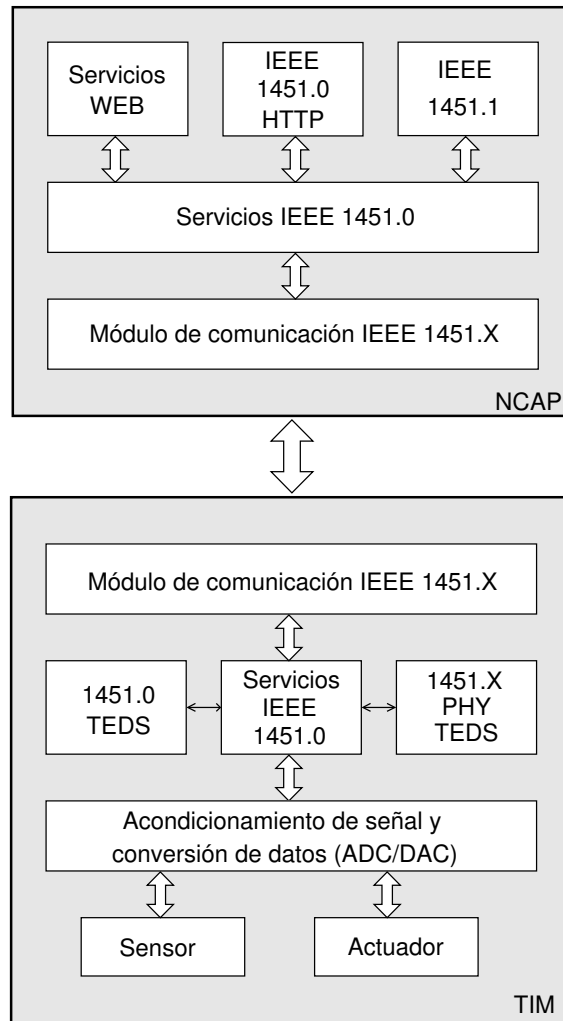


Figura 1.1: Arquitectura del conjunto de estándares IEEE 1451

Entre sus principales problemas se encuentran su complejidad y la baja adopción del IEEE 1451.2 original (TII). Tampoco ha habido demasiado interés en las normas 1451.1 y 1451.3, siendo 1451.4 la excepción en ciertas áreas. Por otro lado, la preparación manual de los TEDS (hojas de datos del transductor) no es sencilla, haciéndose necesario una herramienta para llevar a cabo esta función. Las dos normas clave del estándar, IEEE 1451.0 y 1451.5, han sido recientemente adoptadas. Ambas han sido preparadas más cuidadosamente y, por tanto, es de esperar que su adopción sea más probable. En la Tabla 1.1 se muestran todos los miembros de la familia, su ámbito de aplicación y su estado actual.

1.1 Historia

En el mes de septiembre de 1993, el *National Institute of Standards and Technology* (NIST) y el *Institute of Electrical and Electronic Engineers* (IEEE) promovieron una reunión para debatir acerca de las tecnologías existentes en el campo de los transductores inteligentes. En esta reunión se reconoció la necesidad de crear un interfaz común de comunicación para transductores inteligentes, para lo que se crearon inicialmente cuatro grupos de trabajo (P1451.1, P1451.2, P1451.3 y P1451.4), añadiéndose otros posteriormente.

Norma	Estado
1451.0 TEDS y Comandos Comunes	Aprobado (2007)
1451.1 Interfaz NCAP - Computador	Aprobado (1999) - Requiere revisión
1451.2 Interfaz punto a punto (TII)	Aprobado (1997) - Requiere revisión
1451.3 Buses Multi-Drop	Aprobado (2003) - Requiere revisión
1451.4 Interfaz Analógico	Aprobado (2004)
1451.5 Wireless (WiFi, Zigbee, etc)	Aprobado (2007)
1451.6 CAN Bus	Pendiente
1451.7 RFID	Pendiente

Tabla 1.1: Familia IEEE 1451.0

Desde el punto de vista cronológico, los trabajos evolucionaron de la siguiente forma:

- 31 de marzo de 1993: idea de crear un interfaz común para transductores inteligentes.
- 31 de marzo y 1 de abril de 1994: primera conferencia promovida por el NIST y el IEEE para normalizar el interfaz.
- 19 de septiembre de 1994: primera demostración de un TEDS en la feria Sensors Expo de Cleveland.
- 8 y 19 de marzo de 1995: presentación de la idea de NCAP en la feria Sensors Expo de Boston.
- 14 de septiembre de 1995: cuarta reunión en la feria Sensors Expo de Chicago.
- 15 a 17 de noviembre de 1995: demostración de un modelo de programación orientado a objetos para representar un transductor inteligente genérico (norma P1451.1), Gaithersburg.
- 21 de junio de 1996: primera reunión para discutir la norma P1451.3, Gaithersburg.
- 30 de Agosto de 1996: introduccion del primer borrador IEEE 1451.2.
- 28 de febrero de 1997: introducción del primer borrador IEEE 1451.1 para ser analizado.
- 16 de septiembre de 1997: aprobación de la norma IEEE 1451.2-1997.
- 28 de septiembre de 1999: aprobación de la norma IEEE 1451.1-1999.
- 11 de septiembre de 2003: aprobación de la norma IEEE 1451.3-2003.
- 2004: introducción del primer borrador IEEE 1451.5.
- 25 de agosto de 2004: aprobación de la norma IEEE 1451.4-2004.
- 22 de marzo de 2007: aprobación de la norma IEEE 1451.5-2007.
- 23 de marzo de 2007: aprobación de la norma IEEE 1451.0-2007.

1.2 IEEE 1451.0

Esta norma [2] define un conjunto común de comandos, funcionalidades y formatos de TEDS para todos los miembros de la familia IEEE 1451 que hacen uso de un interfaz digital, ayudando a alcanzar interoperabilidad a nivel de datos entre sus distintos representantes. Las aplicaciones IEEE 1451 pueden acceder a los servicios IEEE 1451.0 a través del interfaz de servicios de transductor IEEE 1451.0 existente en el NCAP, que a su vez se comunicará con el TIM a través del módulo de comunicación IEEE 1451.X correspondiente a cada caso. Para la comunicación entre los diferentes sensores o actuadores y el NCAP se tiene un módulo de comunicación similar (nótese la simetría entre TIM y NCAP en este aspecto).

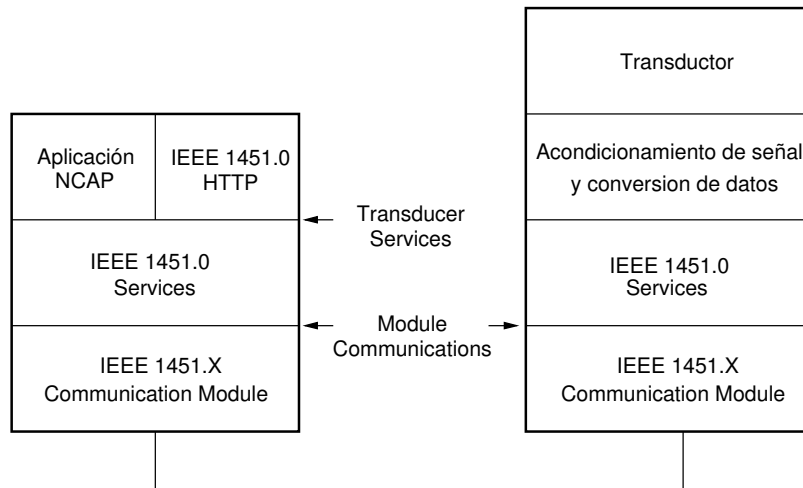


Figura 1.2: Modelo de referencia IEEE 1451

Para que los transductores tengan funcionalidad plug-and-play, tanto el NCAP como el TIM deben cumplir la norma IEEE 1451.0. Sus implementaciones deben soportar las funcionalidades, estructuras de mensajes, comandos y TEDS especificados en la norma, además de uno de los medios físicos IEEE 1451.X y su protocolo de comunicaciones.

Se hace una distinción entre el propio TIM y cada uno de los transductores instalados en éste (denominados *TransducerChannel*). Cada uno de estos transductores es direccionables en grupo o por separado y puede ser de uno de los tres tipos que se describen a continuación.

- **Sensor:** mide parámetros físicos y devuelve una representación digital que representa dicho parámetro. Al recibir un *trigger* (disparo), si el *triggering* está activado, el sensor debe empezar a recibir y almacenar un conjunto de datos en el TIM. El tiempo de cada muestra en el conjunto de datos será controlado por el TIM y es función de los modos de operación del sensor. Un sensor, cuando el transductor está en estado operativo, responderá al comando `Read TransducerChannel` devolviendo el conjunto de datos apropiado.
- **Sensor de eventos:** un sensor de eventos se diferencia de un sensor en que no determina la magnitud de algún fenómeno físico, pero sí es capaz de determinar cuando ocurre un cambio de estado. Este cambio de estado puede ser una señal analógica cruzando un determinado umbral o un conjunto discreto de bits que coinciden con un determinado patrón. Los dos estados posibles son 1 y 0. El modelo de datos de la salida de un sensor de eventos está definido su TEDS correspondiente en el mismo sentido que la salida de cualquier otro sensor.
- **Actuador:** un actuador realiza un determinado cambio físico o provoca una cierta acción. La

salida de un actuador cambia a fin de ajustarse al conjunto de datos cuando ocurre un evento *triggering*. Si hay más de un punto de información en un conjunto de datos, el intervalo entre la emisión del conjunto de puntos de información debe estar bajo el control del TIM.

Cada uno de estos transductores se puede agrupar de dos formas distintas:

- *ControlGroup*: son usados para definir conjuntos de transductores cuando uno de ellos es un canal primario y el resto del grupo o proporciona información adicional sobre éste o son usados para controlar algunos aspectos de éste. Por ejemplo, un *ControlGroup* podría ser usado para definir tres transductores adicionales asociados con un sensor de eventos. Uno de ellos sería un sensor usado para medir la entrada analógica del sensor de eventos; el segundo sería un actuador usado para seleccionar el umbral para el sensor de eventos; y el tercero sería un actuador usado para seleccionar la histéresis para el sensor de eventos.
- *VectorGroup*: se utilizan para definir una relación matemática entre los diferentes transductores. Por ejemplo, podría ser usado para identificar las relaciones entre los componentes de un acelerómetro de tres ejes.

También se definen los denominados *TransducerChannel Proxy*, que se utilizan para combinar la salida de diferentes sensores o la entrada de diferentes actuadores, quizás por razones de eficiencia de transmisión. En cierta forma, son muy similares a un *VectorGroup*.

El estándar define diferentes estados de operación tanto para TIM como para para cada uno de los transductores pertenecientes a éste. En la Figura 1.3 se muestra un diagrama de estados para un transductor. El transductor pasa al estado *Transducer Idle* cuando sale de la inicialización, y puede pasar a estado operativo tras la recepción de un determinado comando, en el que, dependiendo de su configuración, empezará a capturar datos directamente o bien esperará un *trigger*.

Tal y como se muestra en la Figura 1.4 un TIM tiene tres estados. Un TIM permanece en el estado de inicialización por un comando Reset o por un evento de encendido. Una vez que se completa el proceso de inicialización, se hace la transición al estado de TIM activo. Un TIM podrá pasar a TIM sleep por medio del comando TIM Sleep, y el único comando que éste aceptará estando en este estado es Wake-up, que lo devuelve a estado activo.

El formato de un comando entre TIM y NCAP se corresponde con el que se puede ver en la Figura 1.5. Consta de dos octetos para indicar el sensor o actuador de destino dentro del TIM, un grupo de ellos, o todo el TIM; otros dos octetos para indicar la clase de comando de la que se trata y su función; un campo de longitud del mensaje; y una serie de octetos dependientes de cada comando. La clase de comando hace referencia a distintas categorías que se dividen de acuerdo al destinatario del comando y su estado. Por ejemplo, se pueden encontrar clases tales como *Transducer operating state* (para comandos dirigidos a transductores que se encuentran operativos dentro del TIM) o *Tim active state commands* (para comandos dirigidos al TIM cuando está activo). Estos diferentes estados vienen definidos en la norma, tanto para el TIM como para sus diferentes transductores.

En cuanto a la función, se pueden encontrar todas las necesarias. Desde leer los datos de un sensor o desactivarlos, a leer los diferentes TEDS de los que puede constar el TIM. Por ejemplo, si se quieren leer los datos de un sensor, se mandaría un comando tal como el siguiente:

```
DD DD 03 01 LL LL 00 00 00 00
```

DD hace referencia a un transductor en concreto; 03 corresponde al código de la clase de comandos que pueden solicitarse a un transductor en estado operativo; 01 identifica al comando utilizado para leer un sensor, Read TransducerChannel data-set segment; LL es la longitud del mensaje; y los

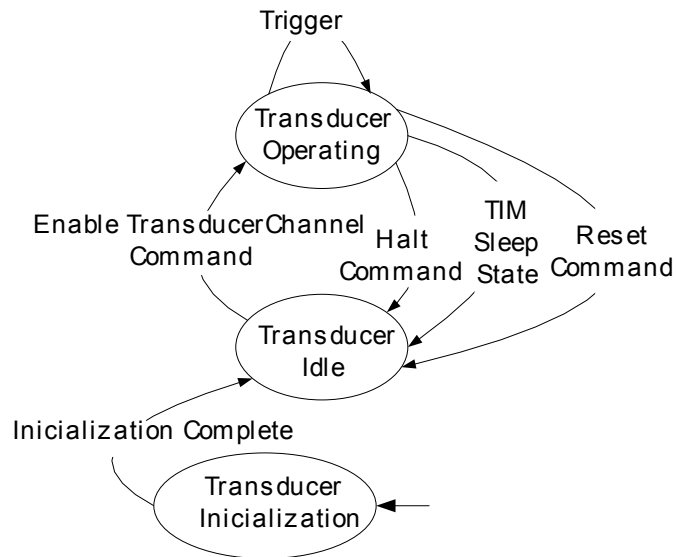


Figura 1.3: Diagrama de estados de un transductor

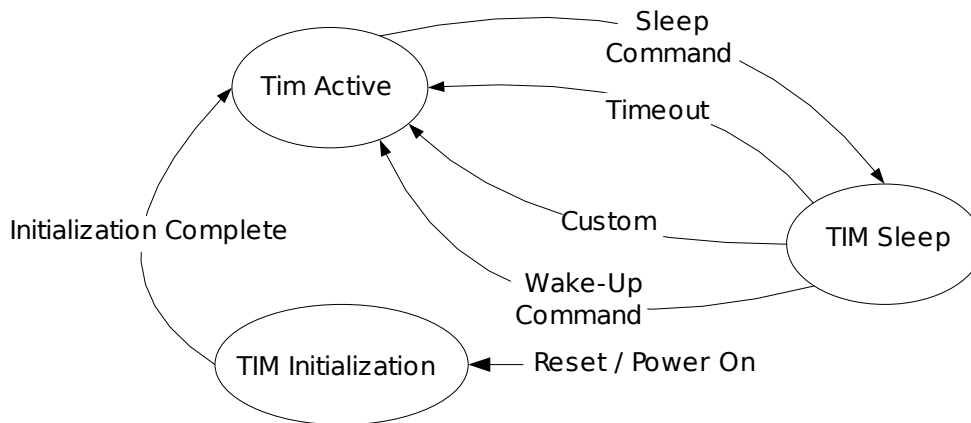


Figura 1.4: Diagrama de estados de un TIM

cuatro últimos ceros corresponden al único argumento que toma este comando (un desplazamiento al leer el conjunto de datos).

Por otro lado, para leer uno de los TEDS se enviaría este comando:

```
DD DD 01 02 LL LL 03 00 00 00 00
```

En este caso, 01 agrupa a los comandos comunes al transductor y al TIM (en cualquier estado) y 02 es el código del comando Read TEDS Segment. Como argumentos, se especifican el código del TEDS a leer (03 es el código asociado al TEDS correspondiente a un transductor concreto del TIM, *TransducerChannel TEDS*) y un desplazamiento como en el caso anterior.

Los mensajes de respuesta deben ajustarse al formato especificado en la Figura 1.6. Así para los dos ejemplos anteriores se obtendrían respuestas como éstas:

```
01 LL LL 00 00 00 00 YY ... YY
```

```
01 LL LL 00 00 00 00 ZZ ... ZZ
```

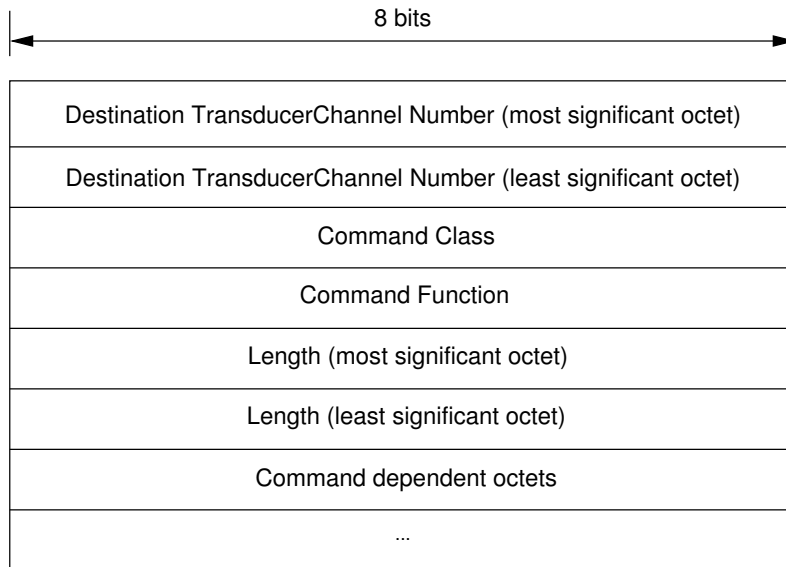


Figura 1.5: Formato mensaje IEEE 1451.0

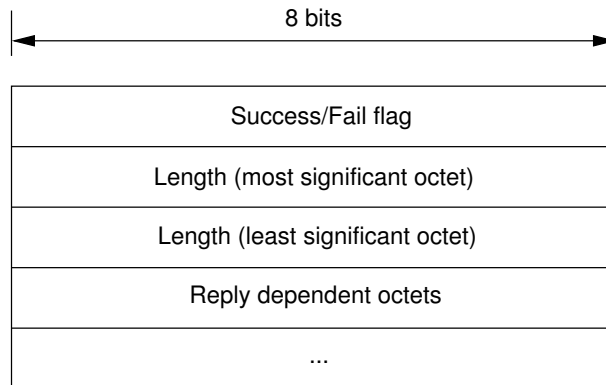


Figura 1.6: Mensaje de respuesta IEEE 1451.0

Nuevamente, los ceros indican un desplazamiento dentro de los datos. Los datos (el TEDS o la lectura del sensor), aparecen al final del mensaje. El formato de cada uno de los diferentes TEDS viene especificado en la norma.

Se supone que estos TEDS deberían ser almacenados en una memoria no volátil del TIM, aunque hay aplicaciones en las cuales esto no resulta del todo práctico. Cuando algún TEDS se almacena en cualquier otro lugar que no sea el TIM, se habla de TEDS virtuales. El fabricante del transductor es responsable de proporcionar un TEDS, pero no de almacenarlo en el TIM. Es responsabilidad del sistema de usuario proporcionar un enlace entre la única información que se garantiza en el TIM, el UUID (*Universal Unique Identifier*), y el archivo del sistema con el contenido del TEDS. La respuesta al comando Query TEDS indica qué TEDS están disponibles y si alguno de ellos es virtual. Es posible que los TEDS sean modificados, bien por la lógica interna de algún transductor, o bien por parte del usuario.

Cuatro TEDS se requieren para cualquier TIM, los demás son opciones. Los requeridos son los siguientes:

- *Meta-TEDS*: ofrece algunos parámetros de tiempo para los peores casos, usados por el NCAP

para determinar si el TIM ha dejado de responder. Además, describe las relaciones entre los distintos traductores existentes en el dispositivo.

- *TransducerChannel TEDS*: da información específica sobre un transductor en concreto, como el parámetro físico que está siendo medido o controlado, el rango en el que se opera, etc.
- *User's Transducer Name TEDS*: proporciona un lugar para que el usuario almacene el nombre por el que el sistema conocerá al transductor. Requiere ser almacenado en una memoria no volátil y que además pueda ser escrito si el usuario lo solicita.
- *PHY TEDS*: depende de la capa física utilizada para conectar TIM y NCAP. No se define en este estándar, aunque sí el método para acceder a él.

Como ejemplo de TEDS opcionales se pueden citar algunos que aportan información sobre localización geográfica, respuesta en frecuencia, etc. Otros TEDS pueden ser definidos por el fabricante. En general, presentan la estructura mostrada en la Tabla 1.2. Tras un primer campo que indica la longitud del TEDS, aparece un bloque de datos que es específico para cada tipo de TEDS. Por último, para proteger la integridad de los datos, aparece una suma de verificación (que también se tiene en cuenta para calcular la longitud del TEDS).

Campo	Descripción	Octetos
--	TEDS length	4
1 a N	Data block	Variable
--	Checksum	2

Tabla 1.2: Formato general para cualquier TEDS

El bloque de datos está formado por conjuntos de datos con una estructura Tipo/Longitud/Valor (denominados TLV). Cada uno de estos campos tiene el siguiente significado:

- El tipo hace referencia a qué es lo que contiene el campo valor. Excepto para 02 y 03, es dependiente de cada tipo de TEDS (por ejemplo, un mismo código no indica el mismo tipo de información en un *Meta-TEDS* que en un *TransducerChannel-TEDS*). Su tamaño es de un octeto.
- La longitud simplemente indica el tamaño del valor correspondiente a cada conjunto TLV. El número de octetos de este campo deberá especificarse en una entrada del conjunto TLV de identificación.
- El valor contiene cualquier información en concreto.

Como ejemplo de TLV, se puede citar el de identificación, el cual es obligatorio en todos los TEDS. Su contenido se muestra en la Tabla 1.3.

En la Tabla 1.4 se ha representado todo el bloque de datos de un *Meta-TEDS* (además del campo de longitud y el checksum). Todos los conjuntos TLV relacionados con la definición de un grupo o un proxy son opcionales.

La Tabla 1.5 muestra el bloque de datos de un *TransducerChannel TEDS*. Por ejemplo, contiene campos como los relacionados con la información de calibración (10, 11), el rango de salida y el error del sensor (13-15) o el modo de muestreo (31, 48, 49). También se puede ver la estructura de un *User's Transducer Name TEDS* en la Tabla 1.6.

Campo	Contenido	Función
Type	03	Identificador para este TLV.
Length	04	Se asume que la longitud de este campo Length es de un octeto, puesto que aún no se ha definido ninguna. Indica que el campo valor (que agrupa a todos los siguientes en esta Tabla) contiene 4 octetos.
Family	00	Identifica al miembro de la familia 1451 que define este TEDS
Class	Clase	Identifica a qué TEDS se está accediendo
Version	Versión	Corresponde al número de versión identificado en el estándar
Tuple Length	Número de octetos	Indica el número de octetos en el campo Length de todos los TLV, excepto éste.

Tabla 1.3: Estructura de identificación de un TEDS

Cada TIM debe incorporar dos tipos de registros de estado. Ambos tienen una longitud de 32 bits. Un registro se denomina *Condition register*, y será leído usando el comando `Read Status-Condition Register`. Contiene el estado actual de los atributos de los que informa. El segundo registro es el *Event register*, y puede ser leído usando el comando `Read Status-Event Register`. Tras ello (o un comando `Clear Status-Event Register`), sus bits pasarán a 0, volviendo a activarse en caso de que alguno de los bits del Condition register pase a estar activo.

Ambos registros de estado serán implementados para el TIM y para cada uno de sus *TransducerChannels*. En muchos casos, un bit en el *Event register* del TIM representa un OR lógico de los bits correspondientes en todos los *TransducerChannels* implementados. Algunos bits son reservados para versiones futuras del estándar. Otros bits son opcionales y podrán ser o no implementados; también se dejan otros abiertos al fabricante. Por último, los que son de carácter obligatorio representan eventos tales como el rechazo de un comando o errores de protocolo.

Este estándar, además de definir los comandos y estructuras de mensajes comentada entre NCAP y TIM, define un par de APIs. Uno de ellos, denominado *Transducer Services API*, es un API utilizado sólo en el NCAP. Este API contiene métodos para leer y escribir transductores, leer y escribir TEDS y enviar comandos de configuración y control a los TIMs. Por ejemplo, para leer datos de un transductor, se utilizaría el método `IEEE1451Dot0::TransducerServices::TransducerAccess::readData`.

El otro API, llamado *Module Communications API*, se sitúa entre el estándar y otros miembros de la familia IEEE 1451. Es un interfaz simétrico que sería implementado tanto en el NCAP como en el TIM, y contiene métodos que serían utilizados por el NCAP para iniciar operaciones de comunicación, de acuerdo a las implementaciones dependientes de cada capa física IEEE 1451.X. Se puede decir que encapsula todos los detalles de la comunicación entre NCAP y TIM.

El objetivo de estos APIs es facilitar el diseño modular para que distintos fabricantes puedan proporcionar distintas funcionalidades e integrarse fácilmente, proporcionando métodos para simplificar la interacción entre aplicaciones de medición y control en el NCAP y el TIM. Los servicios clave son los siguientes:

- descubrimiento de TIMs
- acceso a transductores

Tipo	Nombre	Descripción	Tipo de dato	Octetos
-		Length	UInt32	4
0–2	-	Reserved	-	-
3	TEDSID	Identificación de TEDS	UInt8	4
4	UUID	Globally Unique Identifier	UUID	10
5–9	-	Reservado	-	-
10	OholdOff	Operational time-out	Float32	4
11	SHoldOff	Slow-access time-out	Float32	4
12	TestTime	Self-Test Time	Float32	4
13	MaxChan	Número de TransducerChannels existentes	UInt16	2
14	CGroup	Sub-bloque de información de un Control-Group (junto con tipos 20 y 21)	-	-
20	GrpType	Tipo de ControlGroup	UInt8	1
21	MemList	Lista de miembros del ControlGroup	array de UInt16	NTc
15	VGroup	Sub-bloque de información de un Vector-Group (junto con tipos 20 y 21)	-	-
20	GrpType	Tipo de VectorGroup	UInt8	1
21	MemList	Lista de miembros del VectorGroup	array de UInt16	NTv
16	GeoLoc	VectorGroup especial para localización geográfica (junto con tipos 24, 20 y 21)	--	-
24	LocEnum	Valor que indica cómo se proporciona la información de localización	UInt8	1
20	GrpType	Tipo de VectorGroup	UInt8	1
21	MemList	Lista de miembros del VectorGroup	array de UInt16	NTv
17	Proxies	Sub-bloque de definición de un Transducer-Channel proxy (junto con tipos 22, 32 y 21)	-	-
22	ChanNum	Número asociado al TransducerChannel proxy	UInt16	1
23	Organiz	Organización TransducerChannel proxy	UInt8	1
21	MemList	Lista de miembros del TransducerChannel proxy	array de UInt16	of NT
18–19	-	Reservado	-	-
25–127	-	Reservado	-	-
128–255	-	Abierto a fabricantes	-	-
-		Checksum	UInt16	2

Tabla 1.4: Bloque de datos de un *Meta-TEDS*

- gestión de transductores
- gestión de TEDS

Este API es opcional, la mayor importancia recae en que los mensajes en los interfaces visibles cumplan con el resto del estándar. Es decir, no es necesario, por ejemplo, implementarlo en el TIM si únicamente se quiere que se compatible con cualquier NCAP IEEE 1451.0, y no se desea aportarle mayor flexibilidad a la hora de cualquier tipo de modificación o reutilización.

Más interesante que estas APIs puede resultar el API HTTP definido en el estándar, que estandariza el acceso al NCAP desde la red de usuario a través del protocolo HTTP. Básicamente, ofrece métodos que se corresponden con los ofrecidos por el *Transducer Services API*, con algunas pequeas

Tipo	Nombre	Descripción	Tipo de dato	Octetos
–		TEDS length	UInt32	4
3	TEDSID	TLV de identificación	UInt8	4
10	Calkey	Clave de calibración	UInt8	1
11	ChanType	Tipo de TransducerChannel	UInt8	1
12	PhyUint	Unidades físicas	UINTS	11
50	UintType	Interpretación de diversas unidades físicas	UInt8	1
51-59		Exponente para diversas unidades físicas	UInt8	1
13	LowLimit	Límite inferior operativo	Float32	4
14	HiLimit	Límite superior operativo	Float32	4
15	OError	Incertidumbre peor caso	UInt8	1
16	SelfTest	Clave de auto-chequeo	UInt8	1
18	Sample	–	–	–
40	DatModel	Modelo de datos	UInt8	1
41	ModeLenth	Longitud del modelo de datos	UInt8	1
42	SignBit	Bits significativos del modelo	UInt16	2
20	UpdateT	TransducerChannel update time (tu)	Float32	4
21	WSetupT	TransducerChannel write setup time (tws)	Float32	4
22	RSetupT	TransducerChannel read setup time (trs)	Float32	4
23	Speriod	TransducerChannel sampling period (tsp)	Float32	4
24	WarmUpT	TransducerChannel warm-up time	Float32	4
25	RDelayT	TransducerChannel read delay time (tch)	Float32	4
26	TestTime	TransducerChannel self-test time requisite	Float32	4
31	Sampling	Sampling attribute	–	–
48	SampMode	Sampling mode capability	UInt8	1
49	SDefault	Default sampling mode	UInt8	1
32	DataXmit	Data transmission attribute	UInt8	1
33	Buffered	Buffered attribute	UInt8	1
34	EndOfSet	End-of-data-set operation attribute	UInt8	1
35	EdgeRpt	Edge-to-report attribute	UInt8	1
36	ActHalt	Actuator-halt attribute	UInt8	1
37	Directon	Sensitivity direction	Float32	4
38	DAngles	Direction angles	Dos Float32	8
39	ESOption	Event sensor options	UInt8	1
61–127	–	Reserved	–	–
128–255	–	Abierto a fabricantes	–	–
–		Suma de verificación	UInt16	2

Tabla 1.5: Bloque de datos de un *TransducerChannel TEDS*

Tipo	Nombre	Descripción	Tipo de dato	Octetos
–		TEDS length	UInt32	4
0–2	– Reservado	–	–	–
3	TEDSID	TLV de identificación	UInt8	4
4–9	–	Reservado	–	–
10	Datos del usuario final	Variable	Variable	–
–		Suma de verificación	UInt16	2

Tabla 1.6: Bloque de datos de un *User's Transducer Name TEDS*

diferencias. Un mensaje HTTP de un cliente remoto debe seguir la siguiente sintaxis HTTP URL (RFC2616):

```
http://<host>:<port>/<path>?<parameters>
```

Donde <host> hace referencia a un nodo IEEE1451, <port> al puerto donde escucha, <path> a un comando concreto y <parameters> a sus parámetros. Por ejemplo, así se realizaría la lectura de un sensor:

```
http://129.168.1.1:1451/1451/TransducerAccess/ReadData?ncapId=1&timId=4&channelId=1&timeoutSec=10&timeoutNsec=0&samplingMode=5&responseFormat=text
```

Además de un identificador de TIM y de un sensor (channelId) en concreto dentro de éste, también se puede especificar el formato en el que se quiere la respuesta (texto ASCII, HTML y XML son las opciones recogidas en el estándar).

En la Tabla 1.7 se muestra una lista con todos los métodos soportados por este API.

Discovery API	TIMDiscovery TransducerDiscovery
Transducer Access API	ReadData StartReadData MeasurementUpdate WriteData StartWriteData
TEDS Manager API	ReadTeds ReadRawTeds WriteTeds WriteRawTeds UpdateTedsCache
Transducer Manager API	SendCommand StartCommand CommandComplete Trigger StartTrigger

Tabla 1.7: Resumen API HTTP IEEE 1451.0

Actualmente el standard IEEE 1451.2-1997 no es compatible con el 1451.0, pero el grupo de trabajo del IEEE 1451.2 está en proceso de adaptarlo para conseguir la compatibilidad deseada. Lo mismo ocurre con el IEEE 1451.3-2003.

1.3 IEEE 1451.1

La norma IEEE 1451.1 [3] fue desarrollada para facilitar la creación de software modular y portable para transductores. Define un modelo de información neutro, que se compone de un conjunto jerárquico de clases que representan los diferentes bloques de un NCAP. Este modelo es independiente de cualquier hardware, por tando debe ser portado a una plataforma específica para ser usado. Esta plataforma puede ser cualquier tipo de microprocesador junto con cualquier entorno de red.

Tal como se puede ver en la Figura 1.7, el modelo comprende treinta y cinco clases, a las que hay que añadir las ochenta y una clases de un modelo adicional de datos para la descripción de estructuras y tipos como *Integer8*, *ObjecTag*, *Argument*, etc.

El principal problema de este modelo es su complejidad. A todas estas clases es necesario añadirles el código necesario para acceder a la red y procesamiento de datos para la aplicación. Como resultado, el código fuente para todo el sistema puede ser demasiado grande, especialmente si hablamos de implementarlo en sistemas muy limitados en memoria. Está siendo revisado para ser adaptado a IEEE 1451.0.

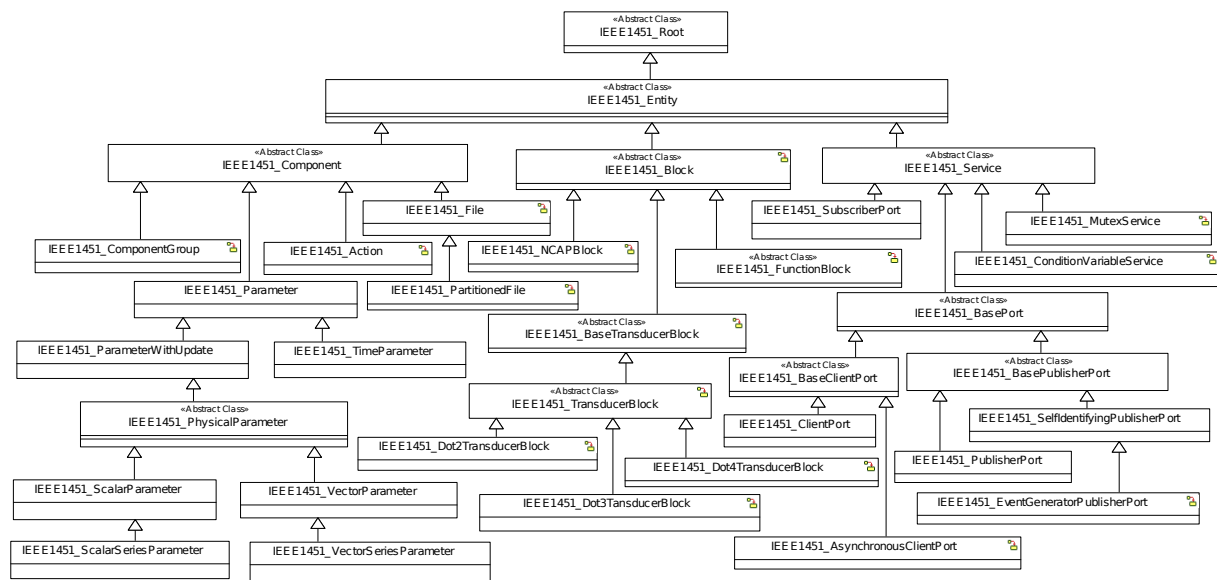


Figura 1.7: Jerarquía de clases del modelo IEEE 1451.1

1.4 IEEE 1451.2

Como ya se ha comentado, esta norma [4] está siendo revisada para cumplir con IEEE 1451.0 y considerar nuevas capas físicas. Originalmente definía una conexión estándar entre TIM y NCAP denominada *Transducer Independent Interface* (TII), un bus de 10 hilos designado para intercambiar información entre ambas entidades. Una razón clave para normalizar el interfaz a nivel de conector de hardware era la necesidad de acabar con los problemas que los fabricantes de transductores se encontraban a la hora de integrar sus productos en distintas redes simultáneamente.

Los TEDS de ese estándar no usan conjuntos TLV como ocurre en el IEEE 1451.0. En cualquier caso, el primer octeto siguiendo al campo *Length* en el *Meta-TEDS* contiene siempre el número 2 (valor reservado en el 1451.0). Ya que este TEDS es el único que contiene información de versión en este estándar, debe ser leído en primer lugar.

Puesto que la mayoría de sistemas comerciales compatibles con IEEE 1451 han dejado de producirse, una desagradable conclusión que se puede extraer es que posiblemente estos productos estaban limitados por el TII, que actuaba como un cuello de botella en lugar de ayudar a un mayor uso del estándar, debido a su complejidad inherente.

Se pueden encontrar ejemplos de implementaciones de IEEE 1451.2-1997 con interfaces físicos distintos al TII, como RS-232 de 3 hilos o USB [5], [6]. Estos interfaces, además de otros, son bastante populares pero no son compatibles con la norma. Por tanto, se ha propuesto que la revisión soporte

RS232, SPI, UART y USB, además del TII. La Figura 1.8 muestra la arquitectura de una red de sensores propuesta basada en IEEE 1451.0 y p1451.2 [7]. Los distintos TIM presentarían una estructura similar al que aparece en la Figura 1.1.

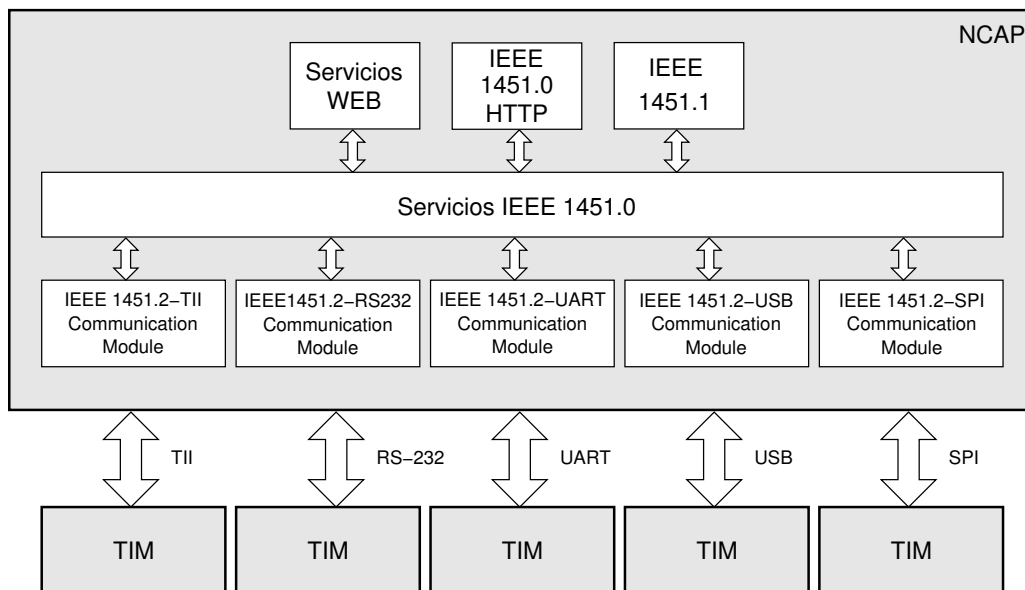


Figura 1.8: Arquitectura propuesta para IEEE 1451.2

1.5 IEEE 1451.3

Este estándar [8] desarrolla un interfaz de transductores inteligentes para para un bus distribuido *multidrop*. El estándar define los interfaces eléctricos, TEDS, protocolos de identificación de canales, protocolos de sincronización y métodos de lectura y escritura para los TEDS y los datos de los transductores. El objetivo es abarcar todos los requerimientos de un sistema distribuido con una gran cantidad de transductores (del orden de cientos) que necesitan ser leídos de forma sincronizada.

No presenta grandes diferencias con el 1451.0, aunque no es compatible por el momento. Por ejemplo, no define los mismos comandos o presenta pequeñas diferencias en el significado de los bits de estado. Al igual que en los dos casos anteriores está siendo revisado para ser adaptado.

En la Figura 1.9 se muestra su modelo de protocolos y su equivalencia con el modelo OSI de 7 capas. Por encima del nivel de enlace, el estándar especifica un determinado formato de datagrama cuya función es servir de transporte para los diferentes comandos y respuestas entre TIM y NCAP; otro para streaming de datos, que es idéntico al anterior salvo en la identificación del protocolo; y otro para la emisión de *triggers* destinados a cualquier transductor o grupo de ellos.

El formato de un mensaje comando, aunque recuerda a los mensajes 1451.0, no es totalmente idéntico, tal como se puede ver en la Figura 1.10. Básicamente, campos que pertenecen a un mensaje 1451.0, como pueden ser la longitud, el tipo de comando o su destino, no aparecen en mensajes de comando 1451.3, situándose esta información en un espacio destinado a esa función dentro del datagrama especificado en el estándar. Algo parecido ocurre para el mensaje de respuesta de la Figura 1.11. Otro detalle importante es que, mientras en IEEE 1451.0 los *triggers* se tratan como cualquier otro comando, en 1451.3 reciben un tratamiento especial, con su propio protocolo situado por encima del nivel de enlace.

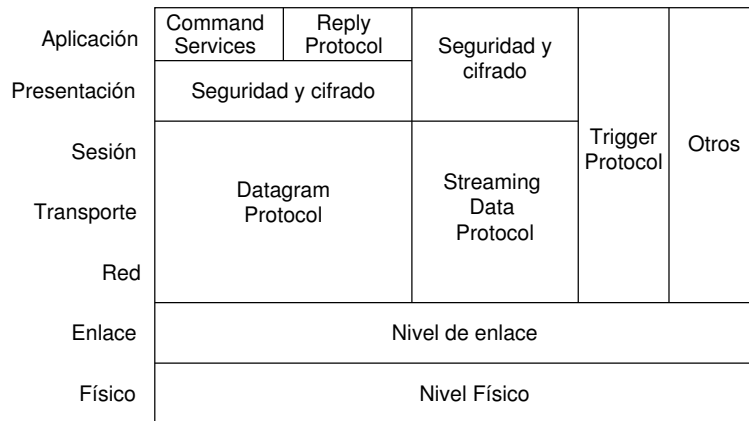


Figura 1.9: Pila de protocolos IEEE 1451.3

Los TEDS de ese estándar tampoco usan estructuras TLV. El primer octeto siguiendo el campo Length en el *Meta-TEDS* contiene siempre el número 1 (tipo reservado en el 1451.0). Ya que este TEDS es el único que contiene información de versión en este estándar, debe ser leído en primer lugar.

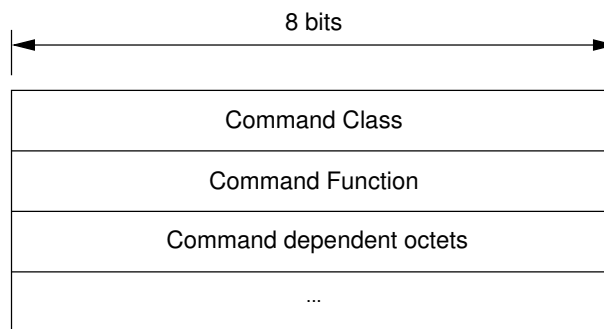


Figura 1.10: Formato de comando IEEE 1451.3

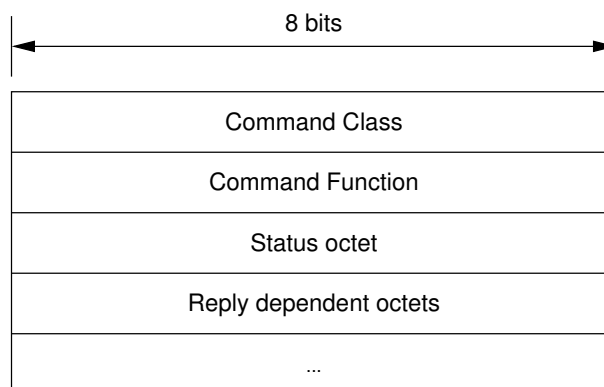


Figura 1.11: Formato de respuesta IEEE 1451.3

También aparece en el estándar una descripción de los servicios que deben ser proporcionados por el nivel de enlace, tales como descubrimientos de los llamados TBIM (*Transducer Bus Interface Module*), los cuales son equivalentes al TIM de IEEE 1451.0, o envío de datagramas. Este nivel de enlace queda dividido en una subcapa LLC (*Logical Link Control*) y una subcapa MAC (*Medium Access*

Control). Para la subcapa LLC se define un formato de trama capaz de albergar los distintos protocolos de la Figura 1.9. La subcapa MAC, responsable del acceso al medio físico, usa un método de contienda con detección de colisiones (CSMA/CD). El formato de la trama MAC es el especificado en la norma IEEE 802.3.

El interfaz físico para el bus de los distintos TBIMs está basado en la especificación *Home Phoneline Networking Alliance* (HomePNA). Un solo par de líneas se utilizan para proporcionar alimentación eléctrica a los transductores y servir de medio de comunicación entre el *Transducer Bus Controller* (TBC) y los TBIMs. En la Figura 1.12 se encuentra ilustrada esta arquitectura.

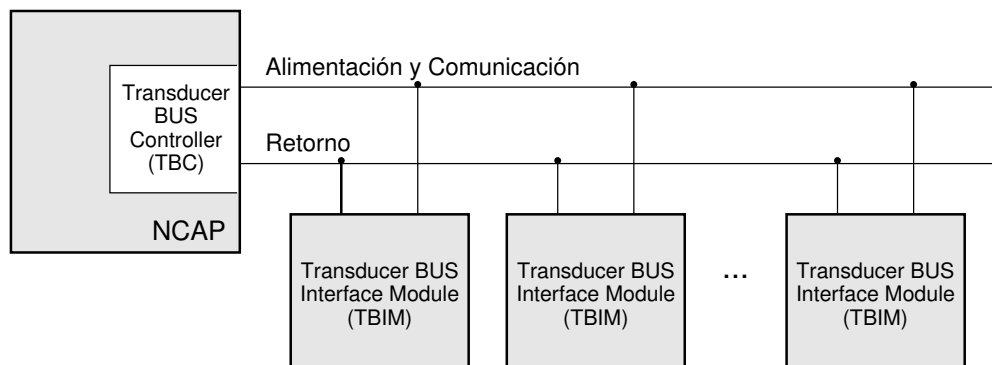


Figura 1.12: Interfaz IEEE 1451.3

1.6 IEEE 1451.4

La norma IEEE 1451.4 [9] propone un interfaz de comunicación normalizado para transductores analógicos. El objetivo es compatibilizar los transductores ya existentes en el mercado con el modelo IEEE 1451, centrándose principalmente en añadir la capacidad de almacenar TEDS a estos antiguos sensores.

Se define el concepto de transductor de modo mixto que proporciona al mismo tiempo tanto un interfaz analógico como uno digital. El interfaz analógico proporciona una señal (tensión, corriente) que representa una magnitud física (temperatura, presión, fuerza, etc.). El interfaz digital se puede utilizar para leer el TEDS y configurar el transductor.

Hay dos tipos de interfaces de modo mixto definidos en el estándar:

- Interfaces de clase 1: definen un esquema para conmutar secuencialmente entre el modo analógico y el modo digital (TEDS) a través de un solo par de hilos.
- Interfaces de clase 2: en este caso se separa el interfaz digital de la salida analógica del sensor. La salida analógica queda intacta, mientras que se añade el interfaz digital en paralelo.

La parte digital en ambos casos es idéntica, estando basada en el protocolo de un hilo de Maxim/-Dallas Semiconductor.

El modelo de TEDS fue modificado para permitir almacenar la mínima información necesaria en pequeños dispositivos de memoria, tal como se requiere para sensores de este tamaño. Por tanto, nada de lo que aparece en el IEEE 1451.0 sobre este aspecto es aplicable aquí. Obviamente, tampoco cumple con dicho estándar en lo referente a funcionalidad y comandos, quedando fuera de su cobertura.

1.7 IEEE 1451.5

El estándar IEEE 1451.5 [10] es un estándar para la comunicación de sensores inalámbricos que define un conjunto de especificaciones para la comunicación entre el *Wireless Transducer Interface Module* (WTIM), equivalente al TIM de IEEE 1451.0, y el NCAP. Adopta los populares 802.11, Bluetooth y ZigBee como sus protocolos de comunicación inalámbrica, aunque podría adoptar otros de ser necesario. Para cada una de estas tecnologías radio se define un PHY TEDS que deberá estar soportado por el WTIM y proporciona una descripción de las funciones y protocolos que debe soportar el módulo de comunicación entre WTIM y NCAP.

Un WTIM se puede describir como un dispositivo que contiene unos de los módulos de radio ya mencionados (*Dot5ApprovedRadio*), acondicionadores de señal, conversores A/D, conversores D/A, sensores y actuadores, aunque no todos necesariamente. Puede llegar a ser tan simple como un sensor o actuador individual más radio o tan complejo como unidades que contienen muchos transductores más radio. Tanto el WTIM como el NCAP contienen un módulo radio similar para completar el enlace de comunicación entre ambos. Existen una serie de especificaciones que hay que tener en cuenta en este caso:

- Un NCAP puede enrutar comandos y datos desde una red externa a/desde un transductor conectado al WTIM.
- Un NCAP puede registrar múltiples WTIMs.
- UN WTIM podrá ser únicamente registrado por un único NCAP.
- Un WTIM puede interactuar con múltiples transductores.
- La comunicación WTIM-WTIM está permitida.

En la Tabla 1.8 se muestra el bloque de datos del PHY-TEDS especificado en esta norma. Como ya se ha comentado, estos TEDS dependen de cada capa física y se especifican en las normas IEEE 1451.X. Especifican características de la capa física inalámbrica como por ejemplo la máxima tasa de transmisión o parámetros de autenticación y cifrado.

Es importante destacar que es la única norma de la familia que cumple plenamente con lo establecido en el IEEE 1451.0. Ha sido realizada teniendo en cuenta los comandos, funciones y TEDS descritos en dicho estándar.

Bluetooth

La sub-especificación Bluetooth proporciona una capa de transporte para implementar el estándar IEEE 1451.0. Su objetivo es definir las funciones, interfaces y protocolos que permiten la comunicación entre TIM y NCAP, soportando Bluetooth 2.0 + EDR o posterior.

La implementación IEEE 1451.5 debe ser responsable de controlar la señalización y el flujo de datos entre NCAP y WTIM. BNEP (*Bluetooth Network Encapsulation Protocol*) es usado para proporcionar capacidad de trabajo en red para los dispositivos Bluetooth sobre L2CAP. SLIP (*Serial Link Internet Protocol*) sirve para convertir entre un flujo de octetos y un flujo de paquetes. RFCOMM proporciona emulación de puertos serie sobre L2CAP. SDP (*Bluetooth Service Discovery Protocol*) se utiliza para el descubrimiento de dispositivos y sus capacidades. *Link Manager*, *Link Controller* y L2CAP son los protocolos Bluetooth correspondientes a las capas 1 y 2 del modelo OSI.

Tipo	Nombre	Descripción	Tipo de dato	Octetos
–		TEDS length	UInt32	4
3	TEDSID	TLV de identificación	UInt8	4
4–9		Reservado		
10	Radio	Tipo de radio	UInt8	1
11	MaxBPS	Máxima tasa de transmisión	UInt32	4
12	MaxCDev	Número máximo de dispositivos conectados	UInt16	2
13	MaxRDev	Número máximo de dispositivos registrados	UInt16	2
14	Encrypt	Cifrado (tipo y longitud de calve)	UInt16	2
15	Authent	Autenticación	Boolean	1
16	MinKeyL	Mínima longitud de clave	UInt16	2
17	MaxKeyL	Máxima longitud de clave	UInt16	2
18	MaxSDU	Tamaño máximo de SDU (payload)	UInt16	2
19	MinALat	Tiempo mínimo para iniciar una transmisión a un dispositivo desconectado	UInt32	4
20	MinTLat	Tiempo mínimo para transmitir la menor carga de datos posible a un dispositivo conectado	UInt32	4
21	MaxXact	Número máximo de comandos en cola	UInt8	1
22	Battery	Indica si el dispositivo tiene baterías	UInt8	1
23	RadioVer	Versión de radio	UInt16	2
24	MaxRetry	Número máximo de reintentos antes de desconectar	UInt16	2
25–31		Reservado		
48–54		Específico para BLuetooth		
48–54		Específico para 802.11		
48–54		Específico para ZigBee		
48–54		Específico para 6LoWPAN		
128-255		Abierto a fabricantes		
–		Suma de verificación	UInt16	2

Tabla 1.8: Bloque de datos de un *PHY-TEDS*

BNEP y RFCOMM se usan como dos capas de transporte alternativas, estando esta última soportada dentro del estándar como un protocolo más antiguo para soluciones punto a punto que está ampliamente soportado en productos comerciales.

1.8 IEEE P1451.6

Esta norma define un interfaz entre transductor y NCAP usando la red de alta velocidad CANopen. Define un mapeado de los TEDs del 1451 a las entradas de diccionario CANopen, así como mensajes de comunicación, procesado de datos, un parámetro de configuración e información de diagnóstico.

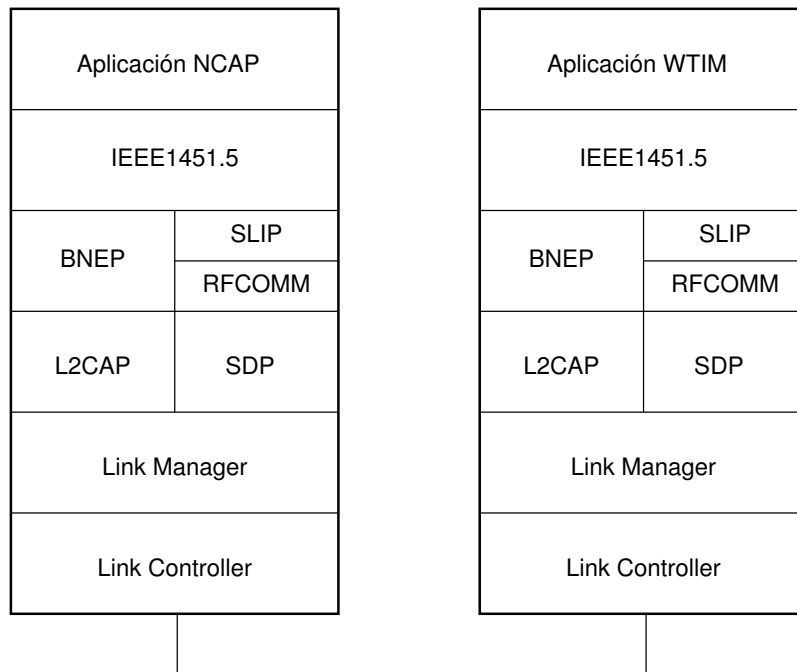


Figura 1.13: Pila de protocolos Bluetooth IEEE 1451.5

1.9 IEEE P1451.7

IEEE P1451.7 [11] define un interfaz y protocolo de comunicación entre transductores y sistemas RFID. Al proporcionar informaciones como la identificación de productos o el seguimiento de su estado, abre nuevas oportunidades tanto para fabricantes de sensores como de sistemas RFID.

1.10 Algunas aplicaciones

En la literatura se pueden encontrar varias aplicaciones del estándar IEEE 1451 orientadas a diversos tipos de redes de sensores. Estas aplicaciones van desde el monitoreo de variables medioambientales, el control del tráfico, sistemas de control, etc. A continuación, se citan algunas de ellas.

Módulos basados en el estándar IEEE 1451 para redes en vehículos inteligentes

En este trabajo [12] se hace un estudio sobre la problemática de la existencia de una gran variedad de tecnologías utilizadas en redes de sensores en vehículos. Se presenta al estándar IEEE 1451 como una posible solución a este problema, ya que el estándar especifica que las funciones de adquisición y conversión de señal en un sensor inteligente deben estar separadas de las funciones de procesamiento y transmisión de esas señales.

Posteriormente se hace un resumen comparativo entre el protocolo CAN, el cual es muy utilizado en las redes vehiculares, y el estándar IEEE 1451 (Figura 1.14), concluyéndose que estas dos tecnologías pueden ser utilizadas conjuntamente para lograr una red de sensores inteligentes de bajo costo de implementación y reparación.

Finalmente se presenta el diseño e implementación de un módulo IEEE 1451.2 basado en el protocolo de comunicaciones CAN. El módulo STIM fue implementado en un PIC 16F877 de la empresa

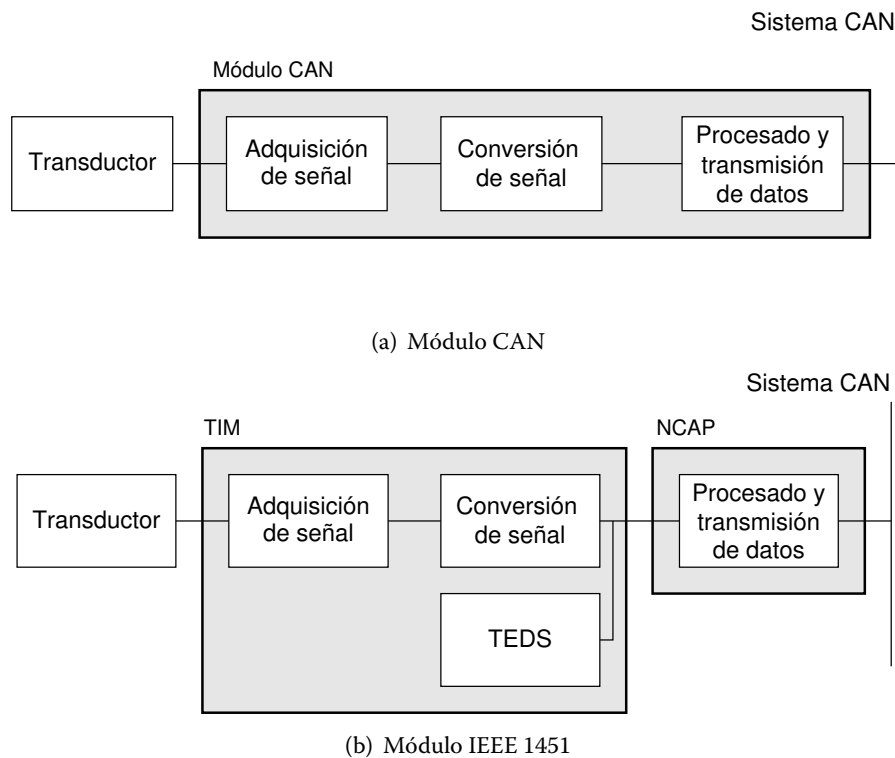


Figura 1.14: Comparativa entre un módulo CAN y otro IEEE 1451

Microchip, mientras que para el módulo NCAP se realizó una implementación en un microcontrolador MC68HC912D60 de Motorola. La comunicación entre el STIM y el NCAP se realizó mediante el TII de 10 hilos. Este artículo fue presentado en la revista *IEEE Transactions on Industrial Electronics* en Diciembre de 2004.

Diseño de una plataforma Plug-and-Play IEEE 1451 para señales de tráfico

Aquí [13] se presenta una aplicación del estándar IEEE 1451 en señales de tráfico. Esta aplicación fue desarrollada utilizando una plataforma de la Universidad de Idaho, que utiliza el micro-controlador Rabbit RCM3000. Además la placa consta de interfaces SPI, CAN, RS232. En la (Fig. 1.15) puede observarse el diagrama funcional de la aplicación de señalización.

Además, se presenta un completo cálculo del tiempo de retardo debido a la implementación del estándar. En la (Fig. 1.16) se puede apreciar un diagrama de tiempo del IEEE 1451.2 en una red distribuida de sensores.

Sistema de control en sistemas multi-frecuencia utilizando el estándar IEEE 1451.0

Aquí [14] se plantea una arquitectura novedosa para el control de sistemas de múltiples frecuencias utilizando el estándar IEEE 1451.0. El STIM fue implementado utilizando una plataforma FPGA al igual que el control del sistema. La arquitectura propuesta fue simulada utilizando una plataforma FPGA y una comunicación serie con un PC compatible.

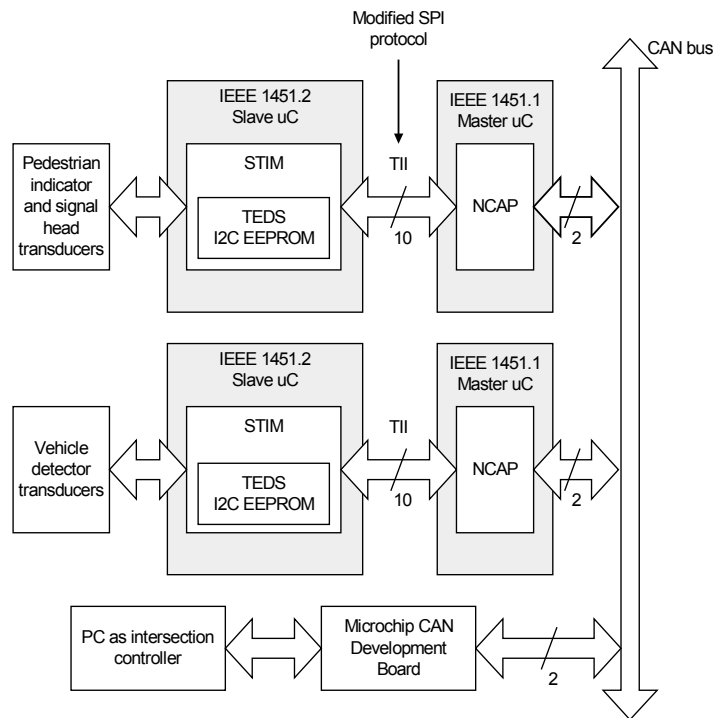


Figura 1.15: Diagrama funcional de la aplicación de señalización

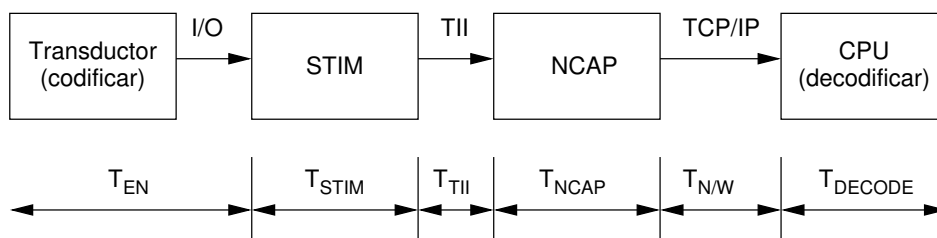


Figura 1.16: Diagrama de tiempo del IEEE 1451.2 en una red distribuida de sensores

Integración de múltiples sensores en sistemas vehiculares utilizando el estándar IEEE 1451: Un caso de estudio

En este trabajo [15] se presenta una integración del estándar IEEE 1451 en el sistema de control de un motor eléctrico para automóviles. Se utiliza un microcontrolador PIC 16F876 para la implementación del STIM, y se utiliza la interfaz TII para la comunicación con el NCAP, el cual es implementado en un PC (Figura 1.17).

Se concluye que el estándar IEEE 1451.2 introduce un retardo de consideración mínima para sistemas no críticos en el automóvil.

Sistema Sensor-Ball basado en IEEE 1451 para la monitorización del estado de las líneas de distribución eléctrica

En este artículo se hace una descripción completa de un sistema de sensores cuyo objetivo es la monitorización de la red de distribución eléctrica. Los elementos clave de todo el sistema son las denominadas *Sensor-Balls*. Cada uno de estos dispositivos hace la función de WTIM y consta de un módulo

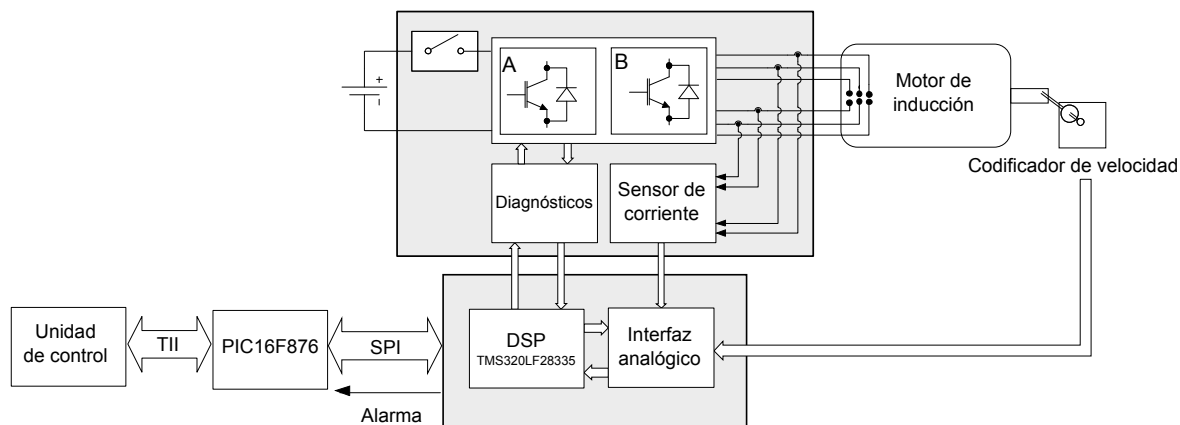


Figura 1.17: Sistema experimental con interfaz IEEE 1451.2 para el control de un motor

de alimentación, un microprocesador que se encarga de su control y un módulo de comunicación inalámbrica (802.11). Sensores de viento (velocidad y dirección), temperatura, corriente, etc. también se encuentran en su interior. Además, existe la posibilidad de instalar otros sensores externos a través de Zigbee.

Para comunicar estos WTIMs con el NCAP se hizo uso de IEEE 1451.0 y 1451.5. Se implementaron los 4 TEDS de carácter obligatorio (*Meta-TEDS*, *TransducerChannel TEDS*, *PHY TEDS* y *User's Transducer Name TEDS*). En el caso del Meta-TEDS se utilizaron algunos de los tipos abiertos al fabricante para definir nuevos campos correspondientes a datos como el número de sensores internos y externos. Para el PHY-TEDS también se definieron dos campos correspondientes a la dirección IP del WTIM y a su dirección de nivel físico.

Una limitación que presenta este desarrollo es que los TEDS de los sensores externos Zigbee se encuentran en el propio WTIM (esto es, en la llamada *Sensor-Ball*), dificultando la sustitución de los mismos. En este caso, no se puede hablar de TEDS virtuales en el sentido que recoge en estándar, debido a que los TEDS no se encuentran en el NCAP.

Referencias

- [1] E. Song and Kang Lee. Smart Transducer Web Services Based on the IEEE 1451.0 Standard. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1--6, May 2007.
- [2] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats. *IEEE Std 1451.0-2007*, pages 1--335, 21 2007.
- [3] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Network Capable Application Processor (NCAP) Information Model. *IEEE Std 1451.1-1999*, pages i--, 2000.
- [4] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. *IEEE Std 1451.2-1997*, pages i--, 1998.
- [5] D. Wobschall and Wai Sing Poh. A smart RTD temperature sensor with a prototype IEEE 1451.2 internet interface. In *Sensors for Industry Conference, 2004. Proceedings the ISA/IEEE*, pages 183--186, 2004.
- [6] H.M. Ramos, P.M. Ramos, and P. Paces. Development of a IEEE 1451 Standard Compliant Smart Transducer Network with Time Synchronization Protocol. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1--6, May 2007.

- [7] E.Y. Song and K.B. Lee. Sensor Network based on IEEE 1451.0 and IEEE p1451.2-RS232. In *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, pages 1728--1733, May 2008.
- [8] 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 Std 1451.3-2003*, pages 0_1--175, 2004.
- [9] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. *IEEE Std 1451.4-2004*, pages 0_1--430, 2004.
- [10] Approved Draft Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. *IEEE Approved Std P1451.5/D10.8, Dec 2006*, pages --, 2007.
- [11] IEEE Draft Standard for a Smart Transducer Interface for Sensors and Actuators - Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats. *IEEE Unapproved Draft Std P1451.7/D1.1, May 2009*, pages --, 2009.
- [12] Kyung Chang Lee, Man Ho Kim, Suk Lee, and Hong Hee Lee. IEEE 1451 based smart module for in-vehicle networking systems of intelligent vehicles. In *Industrial Electronics Society, 2003. IECON '03. The 29th Annual Conference of the IEEE*, volume 2, pages 1796--1801 Vol.2, Nov. 2003.
- [13] R.W. Wall and A. Huska. Design platform for plug-and-play IEEE 1451 traffic signal. In *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*, pages 6 pp.--, Nov. 2005.
- [14] J. Kamala and B. Umamaheswari. IEEE 1451.0 standard based smart readout for multi-rate control of system. *Sensors and Actuators A: Physical*, 154(1):169 -- 174, 2009.
- [15] F. Cortes, F. Barrero, S. Toral, J. Prieto, and J. Guevara. Multi-sensor integration in the vehicular system using the IEEE1451 Std.: A case study. In *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1--6, Oct. 2009.

