

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO



Prototipo *Gateway* IEEE1451-HL7

para servicios *e-Health*

TESIS

que se presenta para cubrir el grado de
MAESTRO EN INGENIERÍA

Raúl Ramírez Ramírez

DIRECTOR DE TESIS:

Juan Iván Nieto Hipólito

Ensenada, B.C.

Julio del 2016.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

Prototipo Gateway IEEE1451-HL7
servicios *e-Health*

TESIS

que para para obtener el grado de maestría en ingeniería

Presenta:

Raúl Ramírez Ramírez

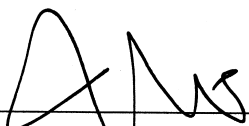
Aprobada por:



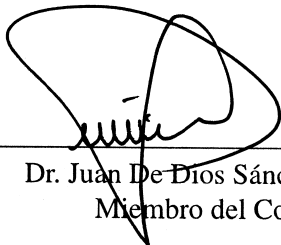
Dr. Juan Iván Nieto Hipólito.
Director de Tesis.



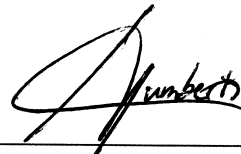
Dra. María de los Angeles Cosío León.
Miembro del Comité.



Dra. Mabel Vázquez Briseño.
Miembro del Comité.



Dr. Juan De Dios Sánchez López.
Miembro del Comité.



Dr. Humberto Cervantes De Ávila.
Miembro del Comité.

Ensenada, Baja California, Julio del 2016.

Resumen de la tesis de **Raúl Ramírez Ramírez**, presentada como requisito parcial para la obtención del grado de MAESTRO EN INGENIERÍA del programa de Maestría y Doctorado en Ciencias e Ingeniería (MYDCI) de la UABC. Ensenada Baja California, México, Julio del 2016.

Prototipo Gateway IEEE1451-HL7
para servicios *e-Health*.

Resumen Aprobado por:



Dr. Juan Iván Nieto Hipólito.

El desarrollo de dispositivos para interconectar diferentes tecnologías es la base para alcanzar la interoperabilidad. A pesar de los avances en los servicios de nacionalización a distancia de Salud Electrónica ó *e-Health*, presentan retos por superar, uno de ellos es la interoperabilidad semántica y sintáctica. El objetivo de este trabajo está enfocado a cubrir esa interoperabilidad, diseñando un *Gateway* ó puerta de enlace, que capture los datos de sensores fisiológicos y los envíe a la nube informática. El dispositivo emplea el estándar de conectividad IEEE 802.15.4 del lado de la red de sensores y el estándar IEEE 802.11 para el envío de la información a la nube. Además, el *Gateway* cumplirá con los estándares IEEE 1451.0, IEEE 1451.1, IEEE 1451.5 y *Health Level 7* con lo que se pretende demostrar la interoperabilidad de la información en el área clínica. Se proporciona un diagrama a bloques para su descripción e interrelación y especificaciones técnicas del *Gateway*. El prototipo *Gateway* fue realizado con *kits* de desarrollo de la compañía *Freescale Semiconductors*. El módulo TWR-K60F120 se usó para cumplir con el estándar IEEE 802.11. Y los módulos MC1322x-LPB y MC1322x-NCB para la red de sensores inalámbricos. Concluimos con retos, nuevas opciones de tecnología de micro-controladores y áreas de aplicación del estándar IEEE 1451.

Palabras Clave: *IEEE 1451, Gateway, HL7, IEEE 802.15.4*

Dedicatoria

A las tres mujeres más importantes en mi vida:

Mi madre (q.e.p.d), *sei sempre nei miei pensieri.*

Mi esposa, *con l'amore.*

Mi hija, *sei la gioia della mia vita.*

Agradecimientos

A mi director de tesis, Dr. Juan Iván Nieto Hipólito, por su invitación a formar parte de este programa, por su dirección y haberme inmerso en este mundo del internet de las cosas.

A mi co-directora de tesis, Dra. María de los Ángeles Cosío León, quién me estuvo guiando y trabajando desinteresadamente conmigo.

A mí comité de tesis por su valioso tiempo, observaciones y sugerencias.

A la *quasi* Dra. Dolores Ojeda Carreño y al *quasi* Dr. José Ramírez, por su compañerismo y su valiosa ayuda técnica.

A la Lic. Guadalupe Haydeé Verdugo Meza y a la C. María Martha Vega Guerrero por su apoyo logístico en su área correspondiente. A todos aquellos que directa o indirectamente han estado conmigo a lo largo de este proceso.

Índice general

Índice de figuras	XV
-------------------	----

Índice de tablas	XVII
------------------	------

1. Introducción	1
1.1. Motivación y pregunta de investigación	3
1.2. Antecedentes	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Limitaciones	5
1.5. Estructura de la tesis	5
2. Prototipo <i>Gateway</i>	7
2.1. e-Health	7
2.2. Interoperabilidad	7
2.3. Estándares clínicos	9
2.3.1. Health Level 7 (HL7)	9
2.3.2. ISO/IEEE 11073	9
2.3.3. Estándar IEEE 1451	10
2.4. Estándares inalámbricos	11
2.4.1. IEEE 802.11 WAN TM	11
2.4.2. IEEE 802.15 WPAN TM	11
2.5. Gateway IEEE 802.11 - IEEE 802.15	12
2.5.1. Definición de <i>Gateway</i>	12
2.5.2. Implementación	13
2.5.3. Trabajo relacionado	14

3. Estándar IEEE 1451	17
3.1. Estándar IEEE 1451.0	20
3.1.1. Hoja de Datos Electrónicos del Transductor - TEDS	20
3.1.2. TEDS requeridos y opcionales	20
3.2. Estándar IEEE 1451.5	22
3.2.1. Estándar IEEE 1451.5 - ZigBee	23
3.2.2. Estándar IEEE 802.15.4	25
3.3. Estándar IEEE 1451.1	28
4. Bibliotecas abiertas IEEE 1451	29
4.1. Biblioteca de Jan Popelka [BJK]	30
4.1.1. TEDS	30
4.1.2. Estructura de datos TEDS_manager	33
4.1.3. Intercambio de TEDS	33
4.1.4. Cómo utilizar la biblioteca	35
4.1.5. Implementación	37
4.1.6. Conclusiones	37
5. Implementación del prototipo Gateway	39
5.1. Descripción del <i>Hardware</i>	39
5.1.1. Módulo MC1322x-LPB	41
5.1.2. Módulo MC1322X-NCB	41
5.1.3. Sistema de torre TWR-K60	42
5.2. Descripción herramientas de <i>Software</i>	43
5.2.1. BeeKit toolkit	43
5.2.2. IAR Systems	45
5.2.3. Teraterm	47
5.3. Aplicación <i>MyStarNetwork Demo</i>	47
5.3.1. MacStandalone	48
5.3.2. MacApps	48
5.3.3. MyStarNetwork Demo	49
5.3.4. MyStarNetworkDemoCoordinator	49
5.3.5. MyStarNetworkDemoEndDevice	49
5.3.6. Exportando el proyecto	50
5.4. IEEE 1451 en IAR	51
5.4.1. Importar el proyecto BeeKit en IAR	51

5.4.2.	Agregar los archivos al proyecto	52
5.4.3.	Configuración TIM y NCAP	53
5.4.4.	Inicialización de TEDS	53
5.4.5.	Envío de TEDS - TIM	53
5.4.6.	Función <i>exchangeTEDSToNCAP()</i>	55
5.4.7.	Recepción de TEDS - NCAP	57
5.5.	Resumen	58
6.	Resultados	59
6.1.	TEDS	59
6.1.1.	Meta-TEDS	59
6.2.	Envío al NCAP	61
6.3.	Recepción del NCAP	61
6.4.	Resumen	63
7.	Conclusiones	65
7.1.	Cumplimiento de los objetivos	66
7.2.	Aportaciones	66
7.3.	Trabajo futuro	67
7.4.	Publicaciones	67
	Nomenclatura	69
	Bibliografía	73

Índice de figuras

1.1. Población Mexicana.	1
1.2. Tecnologías de información	2
1.3. Arquitectura para <i>e-Health</i>	4
2.1. Interoperabilidad para <i>e-Health</i>	8
2.2. Interfaces para <i>e-Health</i>	14
2.3. Interfaces para <i>e-Health</i> tesis	16
3.1. IEEE 1451	19
3.2. IEEE 1451 TEDS	22
3.3. <i>Gateway</i> a <i>WTIM</i>	23
3.4. <i>Gateway</i> a múltiples <i>WTIMs</i>	23
3.5. ZigBee Stack.	24
3.6. Topologías	27
4.1. Estructura de datos TEDS.	32
4.2. Protocolos entre <i>NCAP</i> y <i>TIM</i>	36
4.3. Interfaz IEEE 1451.x - Aeroespacial.	37
5.1. Implementación propuesta.	40
5.2. MC1322x-LPB	41
5.3. MC1322x-NCB	42
5.4. Torre Freescale	44
5.5. Ventana principal de <i>BeeKit</i>	44
5.6. IAR Systems.	45
5.7. Ventana de inicio de IAR Systems.	46
5.8. Programador j-link.	47
5.9. Teraterm logotipo y configuración del puerto serie.	47

5.10. Archivos del proyecto MyStarNetworkDemo.	48
5.11. Exportar el proyecto.	51
5.12. IAR con proyecto de BeeKit.	52
5.13. Ventana IAR con IEEE 1451 en el proyecto.	52
5.14. Configuración <i>TIM</i> ó <i>NCAP</i>	53
5.15. Declaración y manejo de TEDS.	54
5.16. Inicialización del Meta TEDS.	54
5.17. Transmisión del Meta TEDS.	55
5.18. Cadena del Meta TEDS.	56
5.19. Recepción del Meta TEDS.	57
5.20. Función TEDSSetTedsId.	58
6.1. Datos del parámetro UUID.	60
6.2. Meta_TEDS TIM al NCAP.	61
6.3. Envío del parámetro ID del Meta-TEDS.	62
6.4. Meta_TEDS en NCAP.	62
6.5. Envío del parámetro ID del Meta-TEDS.	62
6.6. Interfaces IEEE 1451.x - HL7.	63

Índice de tablas

4.1. Archivos TEDS	32
4.2. Funciones principales.	33
4.3. Funciones auxiliares.	33
4.4. Paquete de error.	34
4.5. Valores que regresa exchangeRespFromTIM.	35
6.1. Meta-TEDS	60

Capítulo 1

Introducción

Actualmente, la infraestructura de salud gubernamental mexicana, que atiende la salud de su población, no es suficiente para dar un servicio de calidad a los 119 millones 530 mil 753 habitantes, reportados por el INEGI en el censo 2015 [16], mostrado en la Fig. 1.1. Algunos de los factores que no lo permiten son: el número de hospitales, número de personal médico y número de pacientes que el médico que puede atender diariamente.

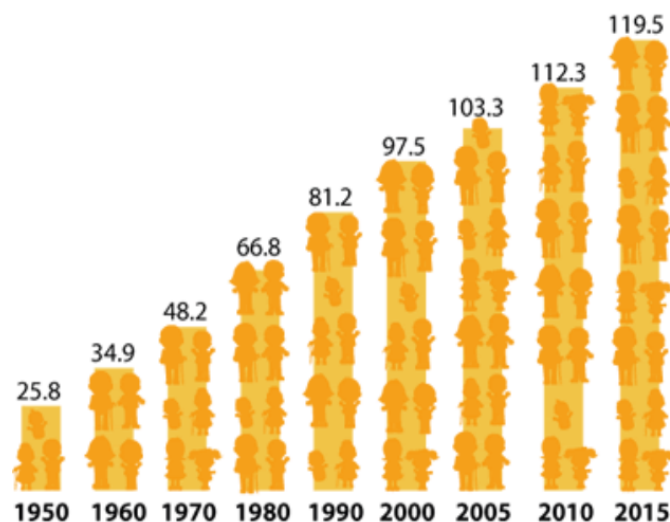


Figura 1.1 Población Total Mexicana 1950-2015.
Millones

Con el desarrollo de la tecnología de la información (IT – por sus siglas en inglés), es posible tener una red informática de salud clínica entre hospitales y establecer una base de datos con información médica de cada paciente. Sin embargo, la lectura y actualización de esta información, se lleva a cabo con sensores tradicionales, por personal médico y

en la sala de los hospitales. Los recientes avances tecnológicos en el rubro de sensores como los circuitos integrados de bajo consumo y comunicación inalámbrica entre otros, han permitido dotar con capacidades y características a los sensores tradicionales, que nos permite llamarlos inteligentes. Si combinamos estos sensores y otros dispositivos electrónicos también denominados inteligentes, como teléfonos, relojes y anillos, nos permite colocar estos sensores en el cuerpo del paciente adscrito y transmitir la información, a través de los diferentes dispositivos inteligentes, a la base de datos del hospital. Con este nuevo esquema, la lectura es tomada y actualizada por los sensores inteligentes incluso en la comodidad del hogar del paciente. Posibilitando al personal médico, incrementar considerablemente el número de pacientes que puede atender, ya que se enfocará en aquellos pacientes que muestren anomalías en parámetros fisiológicos.

Sin embargo, las diferencias entre los protocolos y estándares de comunicación de cada uno de estos elementos, por parte de los fabricantes, hacen que no sean inter-operables *per se*. El presente trabajo busca estandarizar el intercambio de esta información, desde los sensores fisiológicos hasta la base de los datos clínicos, implementando los estándares IEEE 1451 (<https://www2.nist.gov/intelligent-systems-division/ieee-1451-smart-transducer-interface-standards>) y HL7 (<http://www.hl7.org/>) en las tecnologías ya existentes. Con las tecnologías actuales es posible implementar el esquema de La Fig. 1.2 . Donde un grupo de sensores o actuadores pueden conformar en una red de área personal (*PAN* - por sus siglas en inglés) para atender a uno ó más paciente. Estas a su vez conformar una red de área local (*LAN* - por sus siglas en inglés) y poder comunicarse entre hospitales. Por último, las redes locales conformar una red de área amplia (*WAN* - por sus siglas en inglés) para comunicarse entre ciudades, países ó continentes; *id. est.* internet.

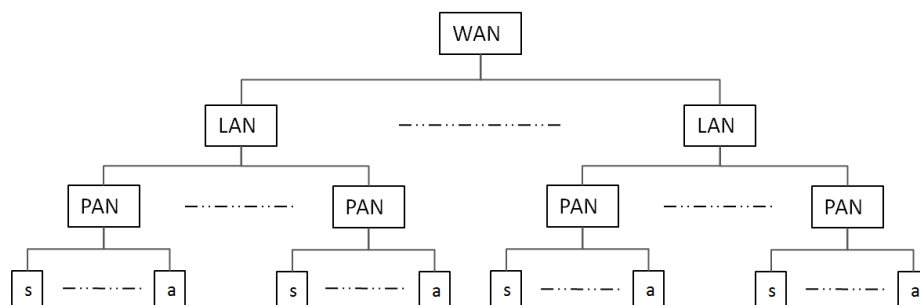


Figura 1.2 Tecnologías de información global.

1.1. Motivación y pregunta de investigación

Habrán 169.5 millones de dispositivos usables ó vestibles para el año 2017 [9]. Los dispositivos electrónicos para el monitoreo de la salud y actividad física en el cuerpo humano, estarán disponibles para la mayoría de la población y a un costo accesible. Una de las aplicaciones de estos productos es la de Monitoreo Remoto del Paciente *RPM* - por sus siglas en inglés). Los *RPMs* son una colección de dispositivos, usables ó vestibles, para la adquisición de datos, los cuáles pueden usar tecnologías de comunicación inalámbrica tales como IEEE802.15.4 (<http://www.ieee802.org/15/pub/TG4.html>), Bluetooth 4.0 (<https://www.bluetooth.com/>), ANT (<https://www.thisisant.com/>) ó Wi-Fi (<http://www.wi-fi.org/>), para el envío, análisis y almacenamiento de la información. La interoperabilidad entre toda esta gama de tecnologías en estos dispositivos, es la principal preocupación hoy en día por ser no compatibles entre ellos, ya sea por las tecnologías mismas ó por violación de los derechos de la tecnología propietarias. Existen grupos y alianzas que hacen un esfuerzo para mantener estas tecnologías interoperables. Algunos de estos grupos y alianzas son ZigBee Alliance (<http://www.zigbee.org/>), Continua Health Alliance (<http://www.continuaalliance.org/>) y Qualcomm Life (<http://www.qualcommmlife.com/>). Los cuales trabajan con desarrolladores de *hardware*, *software* y estándares en cuidados de la salud. Sin embargo, es necesario un dispositivo dedicado, tanto en *hardware* como en *software*, que pueda realizar la integración e interoperabilidad de éstos estándares. Tal dispositivo es llamado *Gateway*, ya que operaría incluyendo la aplicación del usuario.

De lo descrito, surge la pregunta de investigación de éste trabajo de tesis.

¿Cómo disponer de la tecnología de la información y dispositivos inteligentes actuales, para la interoperabilidad de los servicios de *e-Health*?

1.2. Antecedentes

La implementación de la familia del estándar para sensores IEEE 1451, específicamente IEEE 1451.0 [1], IEEE 1451.1 [2] y IEEE 1451.5 [6], en una red de sensores inalámbricos, permiten la interoperabilidad semántica y sintáctica en los estándares comerciales de sensores. Si la información de los sensores en el estándar IEEE 1451.0, a través de un *Gateway*, se agrega el estándar clínico HL7, se tendría la interoperabilidad desde la obtención de las variables fisiológicas de interés hasta el transporte y procesamiento de la información. La Figura 1.3 muestra gráficamente este concepto.

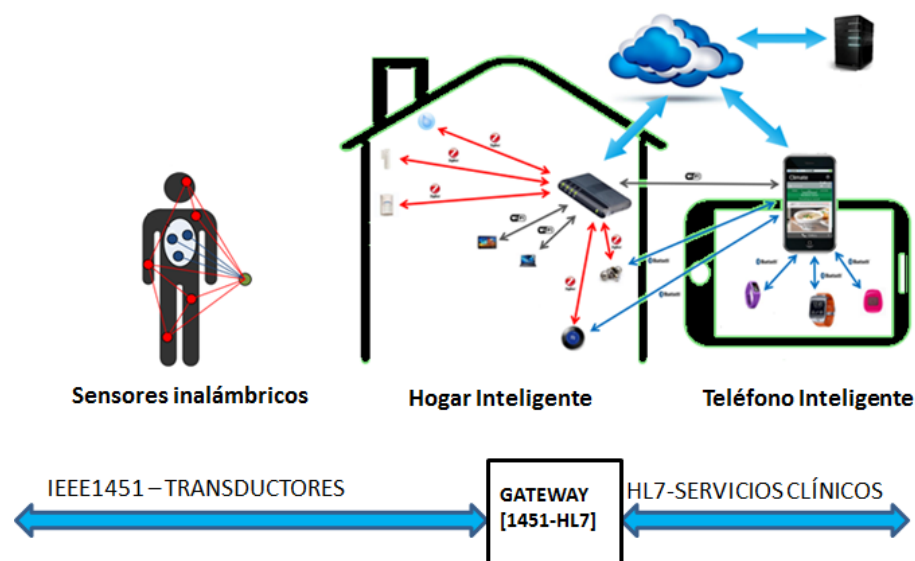


Figura 1.3 Arquitectura de un sistema para la realización de *e-Health*.

1.3. Objetivos

1.3.1. Objetivo general

Diseño e implementación de un *Gateway* prototipo con capacidad de recibir mensajes basados en la familia de estándares IEEE1451.x. y adaptarlos al estándar médico *Health Level 7*.

1.3.2. Objetivos específicos

1. Selección del *Hardware* y ambiente de desarrollo para la implementación del estándar IEEE1451.x en los sensores de variables fisiológicas.
2. Selección del *Hardware* y ambiente de desarrollo para la implementación del *Gateway* IEEE 1451–HL7.
3. Diseño y desarrollo del módulo de conversión de mensajes del estándar IEEE 1451 a la plantilla HL7.

1.4. Limitaciones

El prototipo *Gateway* IEEE 1451-HL7 tendrá las siguientes limitaciones:

1. Se utilizarán *Kits* de desarrollo comerciales de la compañía otrora *Freescale Semiconductors*.
2. En el módulo MC1322x-LPB *Low-Power Board*, se implementará el estándar IEEE 1451.0 para el sensor de temperatura que tiene disponible y el estándar IEEE 802.15.4 para la comunicación inalámbrica.
3. En el módulo MC1322x-NCB *Network Coordinator Board*, se implementará el estándar IEEE1451.0, IEEE 1451.1.
4. Para la comunicación inalámbrica de éstos módulos, se implementará la plantilla generada por el programa BeeKit™ de *Freescale Semiconductors* con la pila IEEE 802.15.4.
5. El puente IEEE 1451-HL7 estará basado en el módulo TWR-K60F120M.
6. Se realizará el *software* indispensable para la prueba de concepto. Esto es, el *software* de configuración del sensor de temperatura y el despliegue de la información para su monitoreo y verificación del estándar.

1.5. Estructura de la tesis

Este trabajo de tesis está estructurado de la manera siguiente: El capítulo 1 es la introducción a la investigación, se describen los antecedentes y la motivación para realizar este proyecto de investigación, los objetivos que persigue y las limitaciones que éste tiene. El capítulo 2 describe los elementos técnicos y de programación requeridos para e-Health, así como la importancia que tiene el *Gateway* en este sistema. El capítulo 3 describe la familia de estándares IEEE 1451 de manera general, pero profundiza en IEEE 1451.0, IEEE 1451.1 e IEEE1451.5 por ser los estándares que se tomaron para éste trabajo. El capítulo 4 describe la biblioteca del estándar IEEE 1451 que se usó en la implementación de éste trabajo. El capítulo 5 describe los módulos, las herramientas de programación y los pasos para la implementación de la biblioteca descrita en el capítulo 4. En el capítulo 6 se presentan los resultados obtenidos. Por último, los logros alcanzados y trabajo a futuro son presentados en el capítulo 7.

Capítulo 2

Prototipo *Gateway*

2.1. e-Health

E-Health es la transferencia de recursos y cuidados de la salud por medios electrónicos. Comprende tres áreas principales:

La disposición de la información medica, ya sea por los médicos ó por los consumidores, a través del Internet y telecomunicaciones. Educa y forma personal de salud, a través de la tecnología y *e-commerce*, con el objetivo de mejorar los servicios de salud pública. El uso de prácticas de *e-commerce* y *e-business* para la gestión de la salud en sistemas informáticos.

E-Health proporciona un nuevo método para emplear los recursos de la salud, tales como la información, dinero y medicinas. Usa el Internet como medio de transmisión de la información. Así como la interacción y colaboración entre las instituciones médicas, profesionales de la salud, proveedores de salud y el público. Con ello, se mejora la calidad de cuidado, la comunicación entre las instituciones médicas, facilita el acceso de la información al personal médico y reduce costos de transportación.

2.2. Interoperabilidad

La industria médica, como cualquier otra industria, ha desarrollado un número de estándares y guías donde no existe regulación alguna. Estos estándares fueron aprobados por industrias líderes para permitir la interoperabilidad entre los dispositivos y para asegurar cierta calidad. El cumplimiento de los estándares no es obligatorio, pero lo hace atractivo al consumidor debido a la interoperabilidad con otros dispositivos. Los estándares pueden tener diferentes propósitos, algunos de ellos, son los siguientes:

- Proveer el criterio de referencia que el producto, proceso o servicio debe cumplir.
- Proveer información que mejore la seguridad, confiabilidad y desempeño de productos, procesos y servicios.
- Garantizar al consumidor la confiabilidad u otras características de bienes o servicios ofrecidos en el mercado.
- Ofrecer al consumidor más productos que permitan que el producto de una empresa pueda ser sustituido por ó combinado con otros productos de otra empresa.

Según Bridget Moorman [22], los agentes involucrados para lograr interoperabilidad en la internet son: dispositivos de red de área personal, área local, área amplia. Así como el programa de aplicación, típicamente implementado en una PC, y los expedientes clínicos, típicamente guardados en una base de datos. La Fig. 2.1 muestra estos agentes.

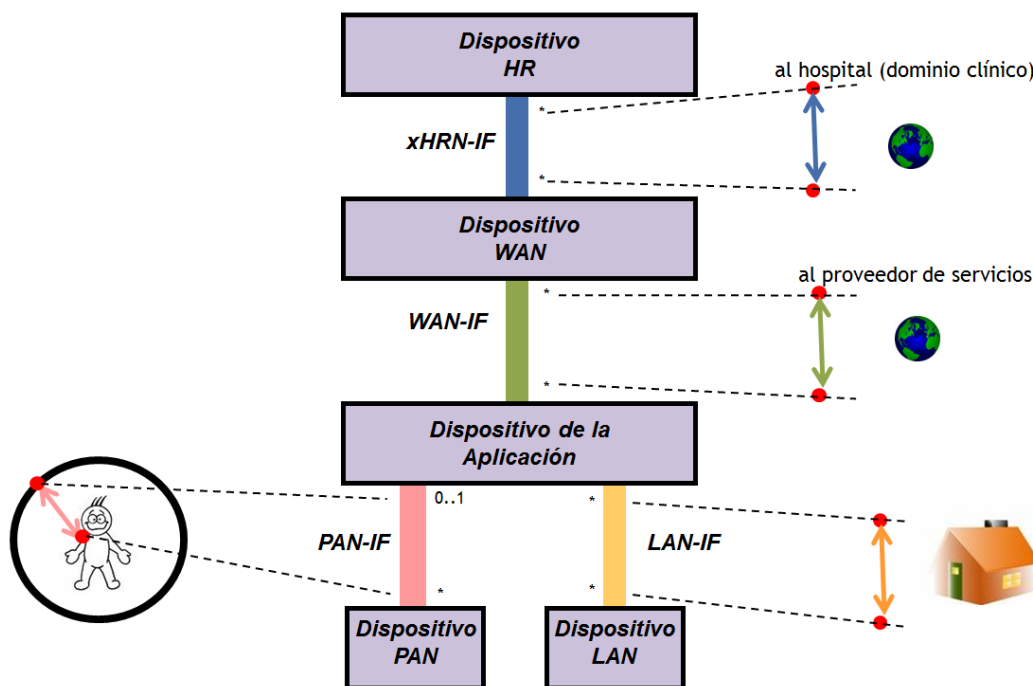


Figura 2.1 Interoperabilidad para *e-Health*, tomada de [22].

En las secciones siguientes se explican las tecnologías y estándares existentes para implementar cada uno de los agentes de esta figura. Así como nuevas opciones para ello.

2.3. Estándares clínicos

Actualmente los estándares clínicos en dispositivos comerciales son Health Level 7 (<http://www.hl7.org/>) y CEN ISO/IEEE 11073. Son explicados en las secciones siguientes.

2.3.1. Health Level 7 (HL7)

Este estándar cumple con el agente Health Record, bloque HR de la Fig. 2.1. Fundada en 1987, Health Level Seven Internacional (HL7) define un marco integral y las normas relacionadas para el intercambio, la integración y recuperación de la información electrónica de salud. HL7 es apoyada por más de 1.600 miembros de más de 50 países, incluyendo más de 500 miembros corporativos que representan a los profesionales sanitarios, las partes interesadas del gobierno, los contribuyentes, empresas farmacéuticas, vendedores / proveedores y empresas de consultoría. Específicamente, HL7 desarrolla los siguientes estándares:

Estándares de mensajes - El estándar HL7 de mensajes v2.x y v3.0 define la manera que los datos son empacados y transferidos de un grupo a otro.

Estándares conceptuales - Los estándares HL7 del modelo de información de referencia (RIM - por sus siglas en inglés), representa los datos clínicos y el ciclo de vida de los mensajes.

Estándares de documento - Los estándares HL7 de arquitectura de documentación clínica (CDA - por sus siglas en inglés), especifica la codificación basada en XML (eXtensible Markup Language), la estructura y semántica de la documentación clínica con el propósito del intercambio de la información.

Estándares de aplicación - El estándar HL7 del grupo de trabajo objeto de contexto clínico (CCOW - sus siglas en inglés), permite al usuario clínico acceder los datos de un paciente en particular a través de múltiples aplicaciones con el uso de un solo nombre de usuario y su clave de acceso.

Para más detalles referirse al trabajo de maestría de Dolores Ojeda [11].

2.3.2. ISO/IEEE 11073

ISO/IEEE 11073 es una familia de estándares adoptado internacionalmente para permitir la completa conectividad entre dispositivos médicos, cuidados de la salud y bienestar. Estos

estándares describen la conectividad desde los niveles físicos hasta la representación abstracta de datos y servicios para intercambiar con computadoras externas. Los estándares están dirigidos hacia los dispositivos de salud personal y de hospitalización. Las metas de los estos estándares es realizar lo siguiente:

- Proveer la interoperabilidad de dispositivos médicos, cuidadores de la salud y bienestar, en tiempo real y *plug-and-play*.
- Facilitar eficientemente el intercambio de los datos de los dispositivos.

Los estándares ISO/IEEE 11073 dispositivo de salud personal (*PHD* - por sus siglas en inglés) son un sub-grupo de la familia de estándares ISO/IEEE 11073, implementan un modelo de comunicación simple debido a que están diseñados para el cuidado personal y no para uso de hospital. Es decir, no accesan a internet. En la siguiente sección se describe el estándar IEEE 1451 para sensores, el cual cuenta con características que permiten el envío de esta información a internet.

2.3.3. Estándar IEEE 1451

El estándar IEEE 1451 (<http://www.nist.gov/el/isd/ieee/ieee1451.cfm>) define una interfaz para transductores inteligentes. 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 el dispositivo transductor, 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. Introduce el concepto de modulo de interfaz del transductor (*TIM* - por sus siglas en inglés) y aplicación para habilitación del proceso de red (*NCAP* - por sus siglas en inglés).

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

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

Las dos normas clave del estándar, IEEE 1451.0 y 1451.5, han sido recientemente adoptadas. Ambas han sido preparadas más cuidadosamente que las anteriores, por lo que se espera una buena adopción por parte de los fabricantes. El resto de la familia de estándares y una explicación más detallada de ellos, se presenta en el capítulo 3.

2.4. Estándares de comunicación inalámbrica

El comité del estándar IEEE 802® LAN/MAN (<http://www.ieee802.org/>) desarrolla y mantiene los estándares de red y recomienda practicas para redes locales, metropolitanas y redes de otras áreas. Usa procesos y acreditaciones abiertas. Los estándares más utilizados son para Ethernet, MANs inalámbricas, RAN inalámbricas, LANs inalámbricas y PAN inalámbricas. En este trabajo, solo describiremos dos estándares, el 802.11 para LANs inalámbricas y el 802.15 para PANs inalámbricas. Estos estándares han sido adoptados para implementar los agentes de las diferentes redes PAN, LAN, WAN indicados en la Fig. 2.1.

2.4.1. IEEE 802.11 WAN™

El estándar IEEE 802.11 WAN (<http://www.ieee802.org/11/>) define el intercambio de información entre sistemas de red local y área metropolitana. El alcance del estándar es la definición de un medio de acceso (MAC - por sus siglas en inglés) y varias especificaciones de capas físicas (PHY - por sus siglas en inglés) para la conectividad inalámbrica para estaciones fijas, portátiles y móviles dentro del área local. El propósito es proveer la conectividad inalámbrica en los dispositivos ya mencionados. Este estándar también ofrece regulaciones para estandarizar el acceso a una ó mas bandas de frecuencia para la comunicación de área local. Los agentes LAN y/o WAN se pueden implementar con el este estándar. IEEE 802.11n tiene una tasa de transferencia de 600 Mbp/s en la capa física. IEEE 1451.5 considera también al 802.11 para WPAN.

2.4.2. IEEE 802.15 WPAN™

El estándar 802.15 WPAN™ (<http://www.ieee802.org/15/>) es para PAN ó redes inalámbricas de corta distancia. Estas WPANs son definidas para dispositivos portátiles y móviles, como PCs, PDAs, teléfonos celulares, entre otros. Permite a estos dispositivos la comunicación, coexistencia e interoperabilidad entre ellos y cualquier otra red (alámbrica ó inalámbrica). Hasta el momento, está conformado por 7 grupos: IEEE 802.15.1, IEEE 802.15.2, IEEE 802.15.3, IEEE 802.15.4, IEEE 802.15.5, IEEE 802.15.6, y IEEE 802.15.7. El agente PAN

se puede implementar con estos estándares. En este trabajo, solo se describirán el IEEE 802.15.1 y el IEEE 802.15.4.

IEEE 802.15.1

Este estándar especifica los requerimientos para información tecnológica, telecomunicaciones e intercambio de información entre sistemas, redes de área local y metropolitana. La meta original de este estándar fue alcanzar un nivel de interoperabilidad que permitiera la transferencia de datos entre los dispositivos WPAN y los dispositivos 802.11TM. Aunque esto resultó ser no viable, el estándar tiene definidos mecanismos que permiten una mejor coexistencia con los dispositivos que cumplen con IEEE 802.11bTM.

IEEE 802.15.4 (LR-WPANs)

Define los niveles de red básicos para redes inalámbricas de área personal con baja tasa de transmisión de datos (LR-WPANs - por sus siglas en inglés), por ende de bajo costo. IEEE 802.15.4 define la capa física y las especificaciones de la sub-capa del control de acceso al medio para lograr la conectividad inalámbrica de baja transferencia de datos, para dispositivos portátiles y móviles que operan sin batería ó un bajo consumo de energía. Operan típica mente en un radio de 10 metros con una tasa de transferencia máxima de 250 Kbp/s (kilo bits por segundo). Este estándar es explicado con más detalle en la sección 3.2.2.

2.5. Gateway IEEE 802.11 - IEEE 802.15

Hasta el momento, hemos explicado y descrito los agentes (*software* y *hardware*) de manera individual que, según Bridget [22], integran la infraestructura e-Health. Sin embargo, no se ha mencionado un elemento primordial para que e-Health funcione tal como se ha mencionado. El *Gateway*, éste elemento hace posible la conexión de una red de sensores *ad hoc*, al sub-sistema WAN. Permitiendo la transferencia de la información entre dos ó más tecnologías y estándares de comunicación diferentes. Debido a ésta propiedad, es muy fácil que se convierta en el sub-sistema más débil.

2.5.1. Definición de *Gateway*

El sub-sistema *Gateway*, es responsable de la conexión de los sub-sistemas BAN y PAN, al sub-sistema WAN. Puede estar implementado en un dispositivo móvil como lo es un

PDA ó *smart phone*, por un nodo de sensores, por una laptop ó por un servidor. Algunos de los nombres que se le da a este elemento son: *gateways*, puentes, *aggregators*, estaciones base ó coordinadores. La complejidad de la arquitectura de la red ó el dominio en el que son implementados influye en el término usado para describir la función en particular. A continuación se describen algunos de ellos.

Gateways - Realiza la traducción de protocolos de diferente red informática. Un *gateway* puede operar en cualquiera de las capas de red, puede comunicarse usando más de un protocolo. Para un servidor ó una interfaz de usuario, una PC puede funcionar como tal. En una red de sensores, un *gateway* es responsable de la interfaz de los nodos de sensores a otra red que use un protocolo diferente. Los *Gateways* trabajan en las capas 1-7 del modelo OSI.

Puentes - Conectan dos ó más segmentos de red por medio de la capa de enlace de datos, capa 2 del modelo OSI, para crear una red *aggregate*.

Aggregators - Son nodos *sink*, que capturan los datos crudos de los nodos en la red de sensores y reducen el tamaño de los datos uniendo en uno solo los datos redundantes ó cor relacionándolos. Con ello se reduce el volumen del tráfico de la red y el consumo de energía del sistema.

Estación base - En una WSN, una estación base es un nodo que cuenta con una mayor potencia computacional, mayor consumo de energía y mayores capacidades de comunicación que los otros nodos de sensores. Una estación base típica actúa como un *gateway* entre los nodos de sensores y el usuario, su función es enviar la información de la WSN al servidor.

Coordinador - En una red ZigBee, un nodo coordinador es responsable de gestionar la red de sensores. Específicamente, el coordinador actúa como la capa de red para seleccionar la frecuencia del canal que será usada por la red, inicializa la red y permite que otros dispositivos se unan a la red. El coordinador también puede proporcionar mensajes de ruta, seguridad y otros servicios.

2.5.2. Implementación

Algunas de las opciones de tecnologías inalámbrica para la implementación de los agentes ya mencionados, se muestra en la Fig. 2.2. Donde,

- Cliente - involucra los agentes HR y WAN, así como HL7.
- Servidor - involucra los agentes WAN, APH y LAN, así como HL7.

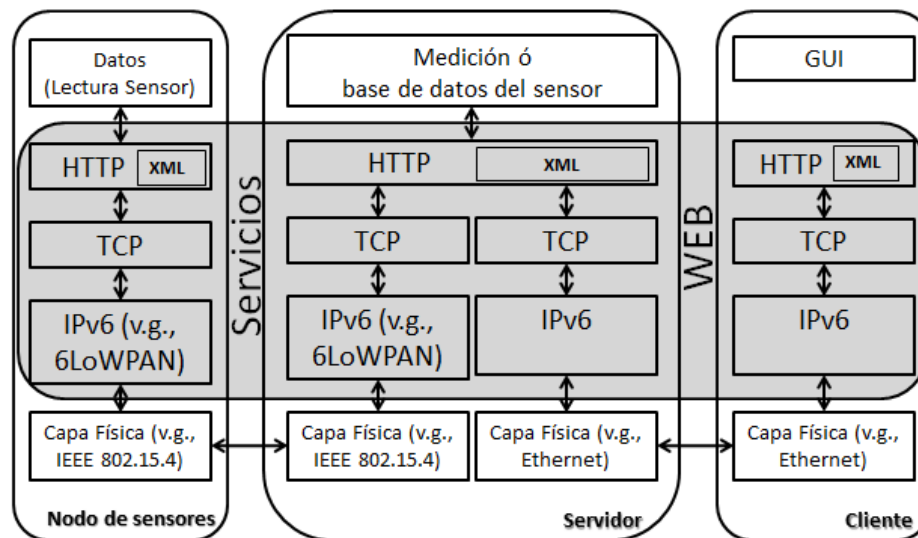


Figura 2.2 Estándares e interfaces para *e-Health*, tomada de [13].

- Nodo de sensores - involucra el agente PAN, así como IEEE 11073.

Recordar que el estándar HL7, especifica la codificación basada en XML, pero no así el estándar IEEE 11073. Por lo que no es posible que HL7 e IEEE 11073 se comuniquen entre ellos por sí mismos. Por otro lado, el estándar IEEE 1451.5 ofrece especificaciones para comunicaciones inalámbricas que cumplen con el estándar IEEE 802.x. y el estándar IEEE 1451.0 especifica la codificación basada en XML. Aunque el estándar IEEE 1451.x no fue diseñado para sensores fisiológicos como el IEEE 11073, es más fácil implementar sensores fisiológicos en el estándar IEEE 1451, que dotar al IEEE 11073 de la codificación basada en XML.

Este trabajo tiene como objetivo implementar un *Gateway* IEEE 1451-HL7 para e-Health en el servidor.

Definiéndolo de la siguiente manera:

- Cliente - involucra los agentes HR y WAN, así como HL7.
- Servidor - involucra los agentes WAN, APH y LAN, así como HL7-IEEE 1451.
- Nodo de sensores - involucra el agente PAN, así como IEEE 1451.

2.5.3. Trabajo relacionado

Esta sección describe algunos de los trabajos relacionados con los estándares IEEE 1451 y HL7 con propósitos clínicos.

Jaeshwan Kim et al [17] hace un estudio para la posibilidad de interoperación entre los estándares IEEE1451 y HL7. Implementa IEEE 1451.2 en el módulo *TIM* y el módulo *NCAP* con un micro-controlador de 8 bits. Hace hincapié en que las características de *plug and play*, disparo de sensor y TEDS del IEEE 1451 son clave para aplicaciones médicas y un gran paso en la solución de interoperabilidad. Sin embargo no implementa los estándares IEEE 1451.0 ni IEEE 1451.5 por estar aún en desarrollo. Y deja para trabajo a futuro la comunicación del *NCAP* con la PC, así como la implementación de la familia de estándares HL7.

Wooshik Kim et al [18] describe la familia de estándares IEEE 1451 y expone como esta familia puede ser usada para los sistemas del cuidado de la salud. Propone agregar los datos del sensor en los atributos de la tabla OBX (segmento de observación), parte del estándar de mensajes de HL7. La implementación del estándar IEEE 1451 la realiza con un asistente digital personal (PDA - por sus siglas en inglés), el cual colecta, procesa y transmite los datos del módulo de sensores.

Marley Lee et al [20] da más detalles técnicos de los estándares IEEE 1451 y HL7. A diferencia de [18], implementa HL7 usando el lenguaje de texto XML. Aunque, al igual que [18], implementa la red de sensores con un PDA, lo hace con el estándar IEEE 802.15.1, mejor conocido como Bluetooth. Concluye diciendo que es difícil hacerlo con sistemas dedicados.

Del análisis de estos trabajos se ve una clara evolución de tecnologías y estándares. Jaeshwan Kim implementa IEEE 1451.2, pero no IEEE 1451.0 ni IEEE 1451.5 por estar en desarrollo en el momento de su trabajo y considera al HL7 para los datos clínicos, pero no lo implementa. Wooshik Kim implementa IEEE 1451.4, IEEE 1451.0 y ZigBee. Ofrece una mecanismo sencillo de comunicación entre los TEDS del estándar IEEE 1451.0 y el estándar HL7, pero no ha sido adoptado por la comunidad clínica. Marley Lee implementa IEEE 802.15.1 y, a diferencia de Wooshik Kim, usa el lenguaje de texto XML para los mensajes clínicos. Lo que permite acercarse a la interoperabilidad de mensajes.

El presente trabajo, pretende integrar los estándares IEEE 1451.0, IEEE 1451.5, ZigBee, IEEE 1451.1 del lado de los sensores y el lenguaje de texto XML, que ofrece HL7, para los mensajes clínicos. E implementarlos en dispositivos con mayores capacidades computacionales y de menor tamaño. La Fig. 2.3 muestra la ubicación de todos estos elementos de acuerdo a la Fig. 2.2. Donde se puede ver que se agregan los estándares IEEE 1451.0 e IEEE 1451.5 que el trabajo de Jaeshwan Kim carece. Es posible implementar cualquiera de los estándares inalámbricos considerados por el IEEE 1451.5, tal es el caso del IEEE 805.15.1 como lo hace Marley. Pero la principal ventaja de éste trabajo es la integración del estándar IEEE 1451.1 que ninguno de los trabajos relacionados lo consideran.

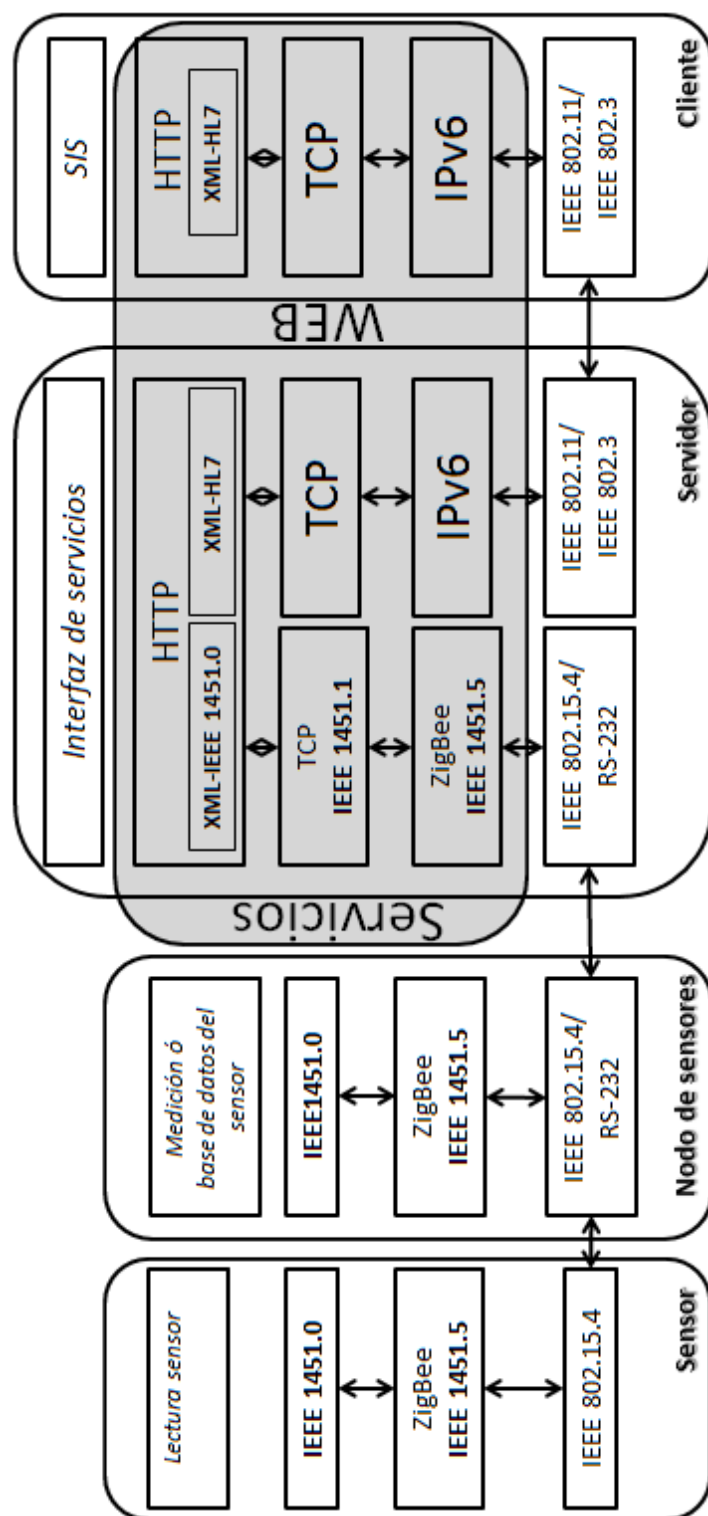


Figura 2.3 Estándares e interfaces para *e-Health* - tesis.

Capítulo 3

Estándar IEEE 1451

La familia de estándares IEEE 1451 es un trabajo arduo para hacer interoperables una gran cantidad de sensores, actuadores, tecnologías de red, procesos y aplicaciones. Define el *Hardware* y *Software* para sensores inteligentes. Provee una interfaz común y habilita la tecnología para la interconectividad de transductores a los microprocesadores, adquisición de datos, sistemas de instrumentación, control y red. La principal característica de este estándar es su implementación en estándares existentes con poca ó nula modificación. Por lo que se reduce tiempo y costo para su implementación. La familia de éste estándar está compuesto por ocho estándares, cada uno define una parte del sistema para dar cumplimiento cabal a la interoperabilidad semántica y sintáctica:

1. IEEE 1451.0 - Define las funciones básicas, protocolos de comunicación y formatos de hojas de datos electrónica (*TEDS* - por sus siglas en inglés), para transductores inteligentes (sensores y actuadores); aprobado en 2007; estado: activo. [1]
2. IEEE 1451.1 - Define el protocolo de comunicación del modelo de información para la aplicación del proceso con capacidad de red (*NCAP* - por sus siglas en inglés) y formatos TEDS; aprobado en 1999; estado: activo. [2]
3. IEEE 1451.2 - Define el protocolo de comunicación del micro-controlador al transductor y formatos TEDS; aprobado en 1997; estado: activo. [3]
4. IEEE 1451.3 - Define el protocolo de comunicación digital y formatos TEDS para un sistema distribuido de caída múltiple; aprobado en 2003; estado: obsoleto. [4]
5. IEEE 1451.4 - Define el protocolo de comunicación y formatos TEDS para sensores análogos a señales de comunicación digital; aprobado en 2004; estado: activo. [5]

6. IEEE 1451.5 - Define los protocolos de comunicación y formatos TEDS para la comunicación inalámbrica de transductores; aprobado en 2007; estado: activo. [6]
7. IEEE 1451.6 - Define los protocolos de comunicación y formatos TEDS para la interfaz de una red de transductores basados en CANopen de alta velocidad; estado: propuesto. [15]
8. IEEE 1451.7 - Define los protocolos de comunicación y formatos TEDS para transductores a sistemas de identificador de radio frecuencia (RFID); aprobado en 2010; estado: sustituido por SRFID-2011. [7]

Esta familia tiene en común formatos, mandos y códigos. Lo que facilita el diseño modular con diferentes funcionalidades, pero mantiene la integración del sistema en uno solo. La Figura 3.1 muestra, de manera general, la disposición de los estándares en el sistema. El módulo interfaz del transductor (*TIM* - por sus siglas en inglés) contiene los *TEDS* con información de calibración y de operación necesarios para presentar los datos en unidades del sistema internacional de medidas SI. Los *TEDS* contiene información adicional para identificar al *TIM* de manera única en un ambiente global. Esta información es transferida al *NCAP* a través de *hardware* y *software* comunes. La parte del *software* cumple con la familia del estándar IEEE 1451 y la parte del *hardware*, para comunicación inalámbrica, puede estar dada por cualquiera de los estándares IEEE 802.11, IEEE 802.15.1, IEEE 802.15.4 y 6LowPAN. Este sistema está centralizado en el *NCAP*, el cual maneja y procesa los datos del *TIM* que serán usados por la aplicación. Cuando el *NCAP* es inicializado, busca *TIM*'s y toma todos aquellos que encuentre. Entonces, transfiere una copia de cada uno de los *TEDS* de estos *TIM*'s a la memoria tipo caché del *NCAP*. Cuando el *NCAP* solicita la lectura del *TIM*, generalmente, éste toma los datos y lo envía en su representación digital básica. El *NCAP* se encarga de realizar, a estos datos, los ajustes indicados en los *TEDS* y los convierte a unidades del sistema internacional. Estos datos son transferidos a una red externa usando protocolo *HTTP* y *XML*. Este mecanismo, permite que los datos de calibración estén asociados al sensor mismo, pero sin la carga compleja de procesamiento de datos. Permitiendo así, la implementación de *TIM* con microprocesadores de 8 ó 16 bits.

Para la implementación de este trabajo sólo se tomaron los estándares IEEE 1451.0, IEEE 1451.1 e IEEE 1451.5. Por lo que, en las siguientes secciones solamente profundizaremos en éstos.

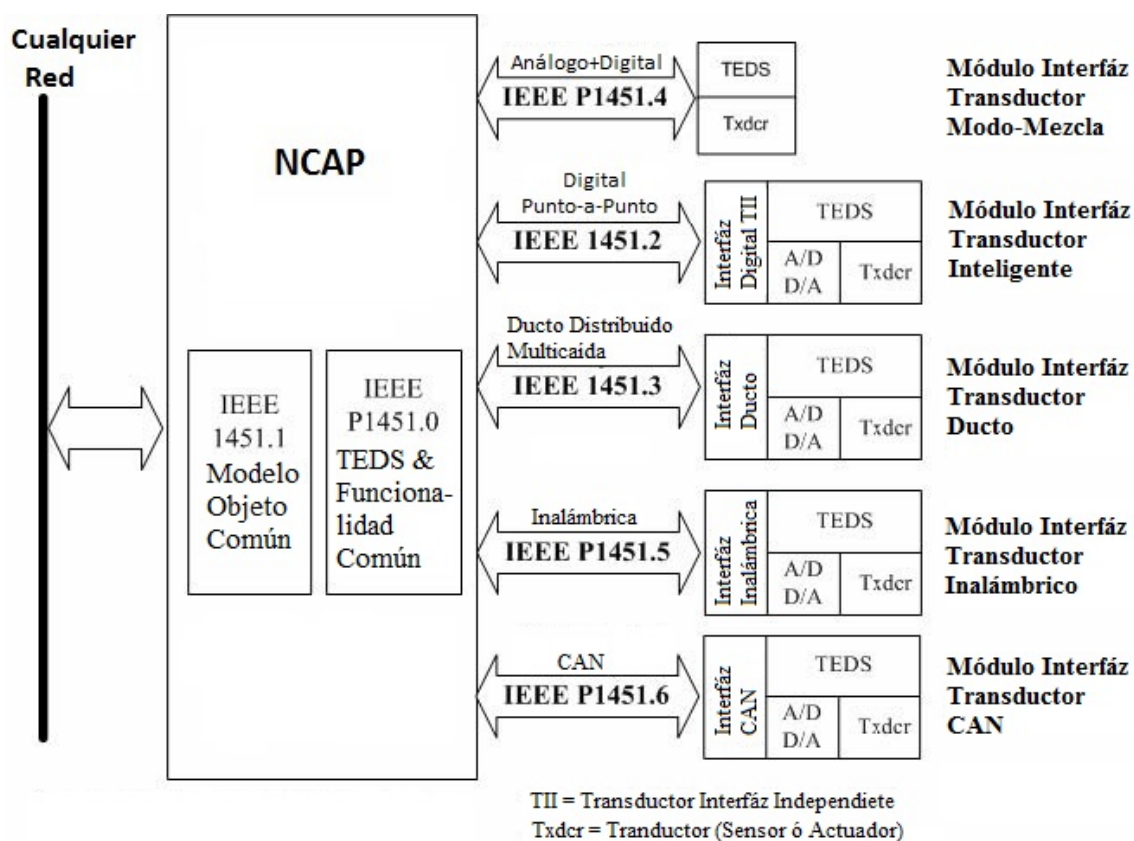


Figura 3.1 IEEE 1451 Estructura General, tomada de [19].

3.1. Estándar IEEE 1451.0

El estándar IEEE 1451.0 [1] provee los elementos básicos para la interoperabilidad del resto de los miembros de la familia de éste estándar. Define las funciones que serán realizadas por el *TIM* y las características comunes de todos los dispositivos que lo implementen, así como las especificaciones de los formatos *TEDS*. También define un grupo de mandos para facilitar la configuración y control del *TIM*, así como la lectura y escritura de los datos usados por éste sistema.

3.1.1. Hoja de Datos Electrónicos del Transductor - TEDS

El concepto de hoja de datos electrónicos del transductor (*TEDS*) es parte muy importante en el estándar IEEE 1451. Aquí se guarda información relacionada con el fabricante y el sensor, como lo son: nombre del fabricante, tipo de sensor, número de serie, calibración. Se consideró ser almacenados dentro del *TIM*, en la memoria no volátil. Sin embargo, hay aplicaciones donde esto no es práctico, así que el estándar permite ser almacenados en otros lugares del sistema. Cuando son almacenados en otro lugar fuera del *TIM*, son referidos como *TEDS* virtuales. Implementar éste concepto permite tener los siguientes beneficios:

- Permite la auto identificación del sensor ó actuador.
- Permite auto documentación a largo plazo - los *TEDS* del sensor puede actualizar y almacenar información como: la ubicación del sensor, fecha de re-calibración, información de reparaciones y mantenimiento.
- Reduce el error humano - hace la transferencia automática de los *TEDS* a la red ó sistema, eliminando así, los posibles errores de ésta información de ser realizada de manera manual.
- Fácil instalación y mantenimiento de los sensores por sus característica denominada "plug and play".

3.1.2. TEDS requeridos y opcionales

La estructura general de los *TEDS*, definida por IEEE 1451.0, especifica que los *TEDS* deberán incluir cuatro *TEDS* requeridos y hasta seis *TEDS* opcionales. La Fig. 3.2, ilustra el formato de los *TEDS* requeridos por la familia IEEE 1451.

Los *TEDS* obligatorios para todos los *TIMs* son los siguientes [10]:

- *Meta TEDS* - Contiene toda la descripción de los TEDS de los transductores conectados al *STIM*. Tales como, parámetros de temporización para casos extremos, los cuales son usados por el *NCAP* para determinar si el *TIM* no está respondiendo. Existe un Meta-TEDS por cada *STIM*.
- *TransducerChannel TEDS* - Contiene la información de unidades físicas, rangos máximos y mínimos, datos del modelo, modelo de calibración y parámetros de disparo, entre otros. El *TransducerChannel TEDS* es específico para cada transductor, por lo que se requiere uno por cada transductor.
- *User's Transducer Name TEDS* - Contiene el nombre del sensor en formato texto, por el cual será reconocido por la aplicación. El estándar recomienda su implementación. Sin embargo, debido a que ésta información es definida por el usuario podría no ser requerida.
- *PHY TEDS* - Éste TEDS es independiente de la comunicación física usada para conectar al *TIM* con el *NCAP*. El IEEE 1451 no lo define, pero define los métodos de acceso.

Por su parte, estos son algunos de los *TEDS* opcionales.

- *Calibración TEDS* - Éste *TEDS* contiene los parámetros de calibración, como lo son fecha y hora de la última calibración, intervalos de calibración, número de segmentos de calibración, los coeficientes polinomiales de calibración, entre otros. Normalmente, la información de éste *TEDS* es específica para cada transductor. Pero, puede darse el caso de aplicar el mismo TEDS a otros transductores.
- *Frequency Response TEDS* - Emplea una tabla para generar la respuesta en frecuencia del *TransducerChannel*.
- *Transfer Function TEDS* - Describe una manera de unir una serie de funciones de transferencia individuales para describir la respuesta en frecuencia de un *TransducerChannel* en forma de algoritmo.
- *Text-based TEDS* - Provee información del *TIM* ó del *TransducerChannel* en forma de texto. Éstos puede estar en uno ó más lenguajes. Consisten en un directorio que facilita el acceso dentro del *TIM*, a un sub-bloque de algún lenguaje en particular, seguido del bloque de *TEDS* que son definidos usando XML.

- *Mandos TEDS* - Son *TEDS* en formato texto que proveen al fabricante la manera de agregar mandos, en caso que se requiera, a los definidos por el estándar. La intención principal de éstos mandos son para el ajuste de transductor y condicionamiento de señal.

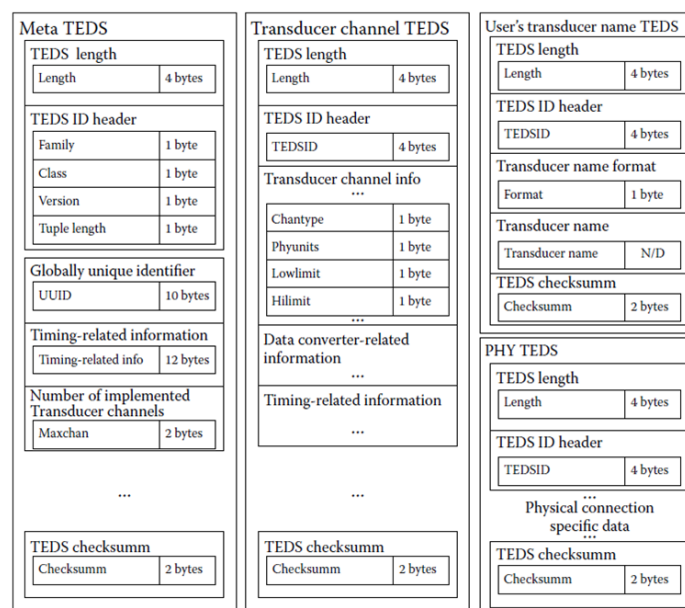


Figura 3.2 Formato de TEDS del IEEE 1451, tomado de [12].

3.2. Estándar IEEE 1451.5

El estándar IEEE 1451.5 [6] introduce el concepto de módulo inalámbrico interfaz del transductor (*WTIM* - por sus siglas en inglés). El cual está conectado por un módulo de radio comunicación aprobado, a un módulo *NCAP*. Los radios aprobados por el estándar IEEE 1451.5 son las tecnologías: IEEE 802.11, 6LoWPAN, IEEE 802.15.1 y ZigBee. Un módulo *WTIM* contiene el acondicionamiento de señal, convertidor análogo-digital y/o digital-análogo y, en la mayoría de los casos, el transductor (sensor ó actuador). El grado de complejidad del *WTIM*, va desde un sólo dispositivo sensor ó actuador con la unidad de radio, hasta varios dispositivos y la unidad de radio. Para completar el enlace de la comunicación inalámbrica entre el *WTIM* y el *NCAP*, se requiere que el *NCAP* cuente con la misma tecnología de radio. Éste estándar se enfoca en los módulos de comunicación para comunicar el *WTIM* y el *NCAP* usando protocolos *Dot5AR*. La función del *NCAP* es enrutar mandos y datos de una red externa a ó desde un transductor conectado a un *WTIM*. Un *NCAP*

puede tener registrados múltiples *WTIMs*. Un *WTIM* sólo estará registrado a un sólo *NCAP*. Un *WTIM* puede tener múltiples transductores; la comunicación entre *WTIMs* es permitida. Las Figuras 3.3 y 3.4 muestran las configuraciones permitidas por éste estándar.

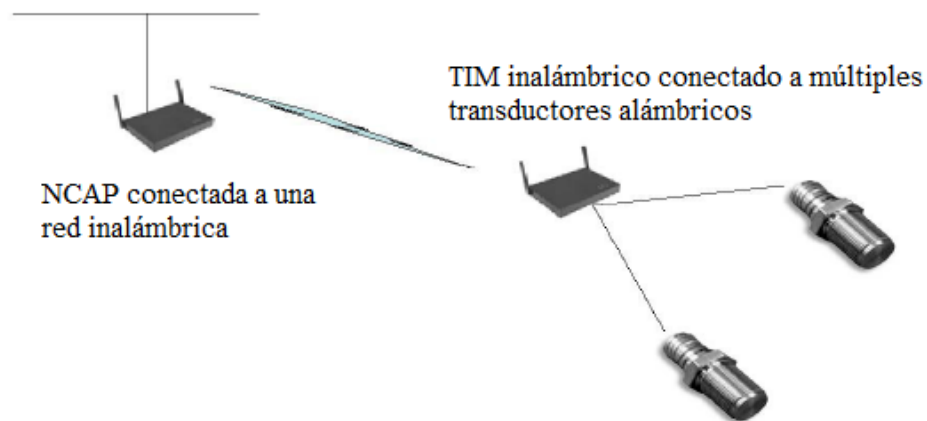


Figura 3.3 Gateway a WTIM.

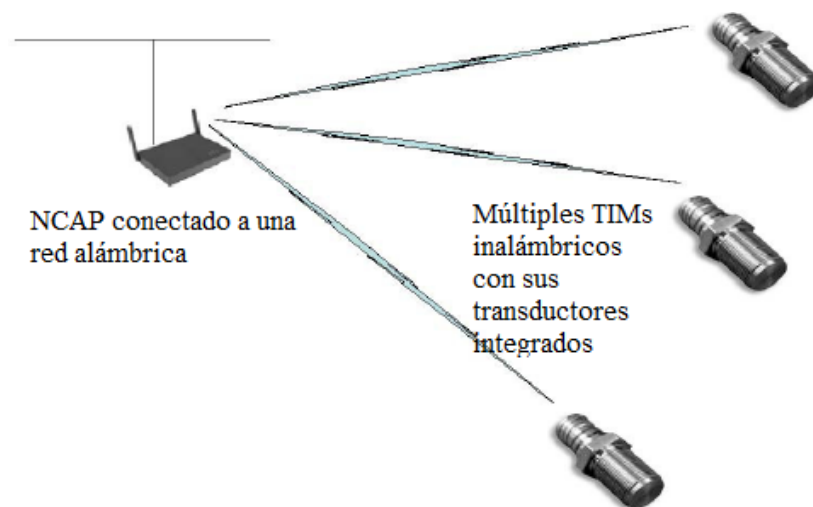


Figura 3.4 Gateway a múltiples WTIMs.

3.2.1. Estándar IEEE 1451.5 - ZigBee

El alcance del estándar de red inalámbrica IEEE 1451.5 - ZigBee, es establecer los requerimientos para implementar una red ZigBee como la capa de transporte en un sistema. Define los perfiles ZigBee de aplicación, temas de seguridad y calidad de servicio. Así como

la capa de convergencia para la interfaz funcional de la red ZigBee con IEEE 1451.0. El protocolo de red ZigBee está definido por ZigBee Alliance (<http://www.zigbee.org/>). La red ZigBee para el sistema IEEE 1451.5 es un intento para definir las funciones, interfaces y protocolos que habiliten la comunicación inalámbrica entre los módulos *NCAP* y *TIM*, por medio del protocolo de red ZigBee. La arquitectura de éste protocolo está organizada de acuerdo al modelo de interconexión de sistemas abiertos (*OSI* - por sus siglas en inglés) y define sólo aquellas capas necesarias para el apropiado funcionamiento, ver Fig. 3.5. La capa física (*PHY*) y la capa de control de acceso al medio (*MAC*) están definidos por el estándar IEEE 802.15.4 [8]. El protocolo ZigBee, toma el trabajo de éste estándar en estas capas como referencia y define las capas de red y aplicación (*APL* - por sus siglas en inglés).

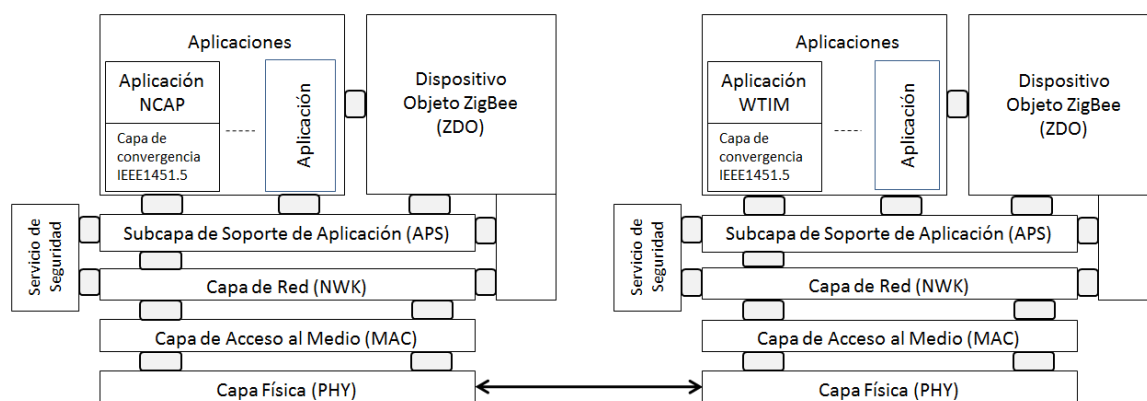


Figura 3.5 ZigBee Stack.

Estructura de la red ZigBee

Una red ZigBee, requiere de la presencia de un coordinador ZigBee así como dispositivos de funcionalidad completa (*FFD* - por sus siglas en inglés) y funcionalidad reducida (*RFD* - por sus siglas en inglés). El dispositivo *FFD* funciona como enrutador, el dispositivo *RFD* sólo como nodo terminal. No es necesario que el módulo *NCAP* con el módulo asociado *WTIM*, se comuniquen directamente ó tener la relación padre/hijo; la capa ZigBee de convergencia mapea los identificadores destinos del IEEE 1451.0 a su correspondiente dirección de red ZigBee, permitiendo que la red ZigBee en-rute el tráfico entre *WTIMs* y *NCAP* asociado sin considerar éste hecho por ninguno de los módulos.

3.2.2. Estándar IEEE 802.15.4

El estándar IEEE 802.15.4 [8] define las redes inalámbricas de área personal de baja transmisión de datos (*LR-WPAN* - por sus siglas en inglés). Las redes de área personal inalámbricas (*WPANs* - por sus siglas en inglés), son empleadas para enviar la información en distancias cortas. A diferencia de las redes inalámbricas de área local (*WLANs* - por sus siglas en inglés), las *WPANs* requieren poca ó nula infraestructura para operar, transferencia de datos confiable, económica y mantiene un protocolo simple y flexible, por lo que es atractivo para una amplia gama de aplicaciones.

Este estándar define la capa física (*PHY* - por sus siglas en inglés) y las especificaciones de la subcapa del control de acceso al medio (*MAC* - por sus siglas en inglés) para la conectividad inalámbrica con poca transferencia de datos en dispositivos fijos, portátiles y móviles con muy poco consumo de energía. Típicamente opera en espacio personal (*POS* - por sus siglas en inglés), aproximadamente en un rango de 10 metros.

Algunas de las capacidades que ofrece éste estándar son:

- Configuración estrella ó punto a punto.
- Dirección única extendida de 64-bit ó única local de 16-bit.
- Asignación garantizada de ranuras de tiempo (*GTSs*).
- Canales de acceso *ALOHA* y *CSMA-CA*.
- Bajo consumo de energía.
- Protocolo con *acuse de recibo*.
- Detector de energía (*ED*).
- Indicador de calidad de enlace (*LQI*).

La capa física considera las siguientes bandas de frecuencia para su transmisión.

- 868–868.6 MHz.
- 902–928 MHz.
- 2400–2483.5 MHz.
- 314–316 MHz, 430–434 MHz y 779–787 MHz para sistemas *LR-WPAN*.
- 950–956 MHz en Japón.

Dispositivos *FFD* y *RFD*

El estándar considera dos tipos diferentes de dispositivos que participan en una red 802.15.4. Dispositivo de funcionalidad completa (*FFD*) y dispositivo de funciones reducidas (*RFD*). Un dispositivo *FFD* puede hacer la función de coordinador de área personal (*PAN*) ó simplemente como coordinador; esta función no la puede llevar a cabo un dispositivo *RFD*. Un *RFD* lleva a cabo funciones donde no se requiere enviar mucha información y está asociado solamente a un dispositivo *FFD*. Ejemplo de éstas funciones son: encender una luz, interruptor, sensar una variable física. Debido a ello, un dispositivo *RFD* puede implementarse usando recursos mínimos y poca capacidad de memoria.

Topología de red

Una red con este estándar, lo conforman básicamente, dos ó más dispositivos que se comunican por el mismo canal físico, donde uno de ellos funciona como coordinador. Pueden formar una topología de tipo estrella ó punto a punto como aparece en la Fig. 3.6. En topología estrella, la comunicación se establece entre los dispositivos y un controlador central, llamado coordinador *PAN*, y serán el punto de inicio ó punto final de la red de comunicación definido en el programa de aplicación. Un coordinador *PAN* también puede tener una aplicación específica, pero deberá iniciar, terminar ó en-rutar la comunicación entorno a ésta red; es controlador primario de la *PAN*. Todos los dispositivos en la red, con cualquier topología cuenta con una dirección única (*extended addresses*) ó dirección corta (*short adress*). El dispositivo usará comunicación de dirección extendida para la comunicación directa dentro de la *PAN* ó dirección corta asignada por el coordinador de la *PAN* al momento de la asociación de los dispositivos. La topología punto a punto también cuenta con un coordinador *PAN*. Sin embargo, difiere de la topología estrella en que cualquier dispositivo es capaz de comunicarse con cualquier otro siempre y cuando se encuentren dentro del rango de alcance. La comunicación punto-punto permite implementar topología de red más complejas como lo es la topología de malla; no se abordará en este trabajo.

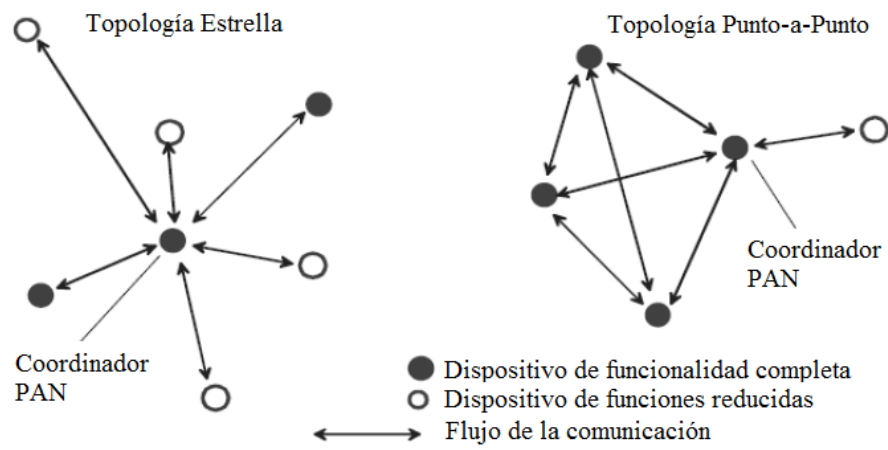


Figura 3.6 topología estrella y punto-a-punto.

3.3. Estándar IEEE 1451.1

El estándar IEEE 1451.1 [2] define una interfaz para conectar procesadores con capacidad de red, con un modelo informático objeto de control de red, para sensores y actuadores inteligentes. Este modelo objeto incluye las definiciones de:

- Bloques para transductores.
- Bloques para funciones.
- Bloques para *NCAP*.

Actualmente existen redes informáticas que permiten a los transductores ser accedidos por medio de ésta red. El propósito de éste estándar es proveer un modelo de aplicación de red neutral que permita reducir los esfuerzos para la interfaz de los sensores y actuadores inteligentes a dicha red.

Un sistema basado en este estándar gozará de los siguientes beneficios:

- Uniformidad en el diseño del modelo para la implementación del sistema.
- Conjunto de operaciones, de red independiente, para la configuración del sistema.
- Define los modelos de comunicación de red independiente.
- Interoperabilidad de todas las comunicaciones.
- Define los modelos de redes independientes para la implementación de la aplicación.
- Modelos de aplicación portátil.

Capítulo 4

Bibliotecas abiertas IEEE 1451

La parte medular de ésta tesis consiste en la implementación del estándar IEEE 1451 en los módulos *TIM* y *NCAP*. Para lo cual se realizó una búsqueda informática, a continuación una breve descripción de los trabajos considerados para la elección y realización en ésta tesis.

1. Implementación libre de IEEE 1451.- La cual ofrece código para sensores inteligentes con servicio web, desarrollada por Eugene Y. Song and Kand Lee, ambos del Instituto Nacional de Estándares y Tecnologías (*NIST* - por sus siglas en inglés) el cual usa ambientes de desarrollo en Java: JDK 1.5.0.10, Netbeans 5.0 y un servidor de web 8.0 de la compañía *SUN* [27].
2. Biblioteca de desarrollo IEEE 1451.- Fue creada para depurar TEDS en una PC y micro-controladores. Puede pasar TEDS en archivos *XML* contenidos en la PC y exportarlos a la memoria en un archivo de cabecera a un programa de sistema dedicado. También puede generar paquetes de mandos, de respuesta y control de comunicación entre el *TIM* y *NCAP*. Trabajo de tesis realizado por Jan Popelka en el año 2010 [23].
3. ADuC812 como transductor IEEE 1451.2.- Nota de aplicación donde se implementa el estándar IEEE 1451.2 para el micro controlador ADuC812 de Analog Devices. Realizado en el año 1999 [26].
4. The PICmicro® MCU as an IEEE 1451.2 Compatible Smart Transducer Interface Module (STIM) - Nota de aplicación donde, se implementa el estándar IEEE 1451.2 para el micro-controlador PIC16C77x de 8-bits [25].

La primera opción ofrece el código fuente en lenguaje Java y el tamaño de memoria requerida para su implementación es de aproximadamente 7 Mbytes. La capacidad de

memoria disponible en el micro-controlador es de 128 Kbytes, por lo que no es posible implementarse en ellos. La tercera opción es código fuente en lenguaje C y requiere tan sólo 8 Kbytes de memoria, para su implementación. Está diseñado exclusivamente para el estándar IEEE 1451.2 con un micro-controlador de 8 bits; ADuc812 fabricado por Analog Devices. La cuarta opción, está implementada en código C, requiere de 52 Kbytes de memoria para su implementación. También está diseñado exclusivamente para el estándar IEEE 1451.2 con un micro-controlador de 8-bits; PIC16C77x fabricado por Microchip. La segunda opción también ofrece el código en lenguaje C para toda la familia del estándar IEEE 1451 y para cualquier micro-controlador. Se tomó ésta opción para implementarla en los *kits* de desarrollo de esta tesis.

4.1. Biblioteca de Jan Popelka [BJK]

La biblioteca de Jan Popelka es una tesis de doctorado que trata del intercambio de datos, entre los módulos *TIM* y *NCAP*, de acuerdo al estándar IEEE 1451 . La biblioteca especifica la interfaz digital para acceder los TEDS, la lectura de datos de los sensores y escritura en los actuadores. Define un conjunto de funciones para lectura y escritura para los TEDS. Con ésta biblioteca se puede obtener capacidad '*plug-and-play*', como lo indica el estándar, de los transductores. Esta biblioteca implementa el estándar IEEE 1451 a través de 12 archivos.

4.1.1. TEDS

BJK diseñó ésta biblioteca para almacenar datos en memoria TEDS, generar una serie de octetos, que contiene datos de la memoria TEDS para la transmisión entre los módulos *TIM* y *NCAP* y almacenamiento de datos de estos campos en la memoria TEDS. Por esta razón, define estructuras de datos que contienen la memoria bloques de datos TEDS y los datos adicionales necesarios para obtener una descripción completa de la memoria y la biblioteca de funciones adecuada y puede reconocer módulo de mandos individuales *NCAP*. La biblioteca consta de varias clases que describen los diferentes bloques de memoria TEDS implementadas e incluye características para su funcionamiento. Una lista de las distintas clases y sus descripciones que contienen están descritos en la Tabla 4.1 La estructura de esta biblioteca se muestra en la Figura 4.1. La clase TEDS es importado por todas las demás clases y la clase TEDS *manager*. Estas clases contienen funciones para la creación de nuevos datos, inicialización de los datos, funciones para la eliminación de estas estructuras de la

memoria, el algoritmo de comprobación de datos *checksum*, cálculo de la longitud de los datos y bit de validez del TEDS. Las funciones para tales tareas son las siguientes:

- TEDSMetaCreate, TEDSPhyCreate, TEDSChanCreate, TEDSCaliCreate, TEDSTranCreate, TEDSFreqCreate, TEDSUserCreate, TEDSNameCreate, TEDSManfCreate, TEDSUnexCreate.
- TEDSMetaInit, TEDSPhyInit, TEDSChanInite, TEDSCaliInit, TEDSTranInit, TEDSFreqInit, TEDSUserInit, TEDSNameInit, TEDSManfInit y TEDSUnexInit.
- TEDSMetaDelete, TEDSPhyDelete, TEDSChanDelete, TEDSCaliDelete, TEDSTranDelete, TEDSFreqDelete, TEDSUserDelete, TEDSNameDelete, TEDSManfDelete y TEDSUnexDelete.
- TEDSMetaGenerateLength, TEDSPhyGenerateLength, TEDSChanGenerateLength, TEDSCaliGenerateLength, TEDSTranGenerateLength, TEDSFreqGenerateLength, TEDSUserGenerateLength, TEDSNameGenerateLength, TEDSManfGenerateLength y TEDSUnexGenerateLength.
- TEDSMetaGenerateChecksum, TEDSPhyGenerateChecksum, TEDSChanGenerateChecksum, TEDSCaliGenerateChecksum, TEDSTranGenerateChecksum, TEDSFreqGenerateChecksum, TEDSUserGenerateChecksum, TEDSNameGenerateChecksum, TEDSManfGenerateChecksum y TEDSUnexGenerateChecksum.
- TEDSMetaSetValid, TEDSPhySetValid, TEDSChanSetValid, TEDSCaliSetValid, TEDSTranSetValid, TEDSFreqSetValid, TEDSUserSetValid, TEDSNameSetValid, TEDSManfSetValid y TEDSUnexSetValid.

Las clases que contienen las hojas de datos individuales incluyen estructura de datos TEDS que describe la hoja de datos. Estas estructuras contienen elementos comunes a todas las hojas de datos: Length, Checksum, Exist, Attrib, y Status.

Tabla 4.1 Archivos TEDS

Archivo	Descripción
TEDS_manager	Clase que administra los TEDS.
TEDS	Clase para la base de TEDS.
TEDS_meta	Clase que describe meta datos.
TEDS_phy	Clase para TEDS físicos.
TEDS_chan	Clase que describe el canal de comunicación.
TEDS_cali	Clase que contiene la calibración.
TEDS_freq	Clase que describe respuesta en frecuencia.
TEDS_tran	Clase que contiene la función de transferencia.
TEDS_name	Clase que describe el nombre del transductor dado por el usuario.
TEDS_unex	Clase que describe las unidades de medida extendidas.
TEDS_manf	Clase que define los TEDS dados por el fabricante.
TEDS_user	Clase que define los TEDS dados por el usuario.

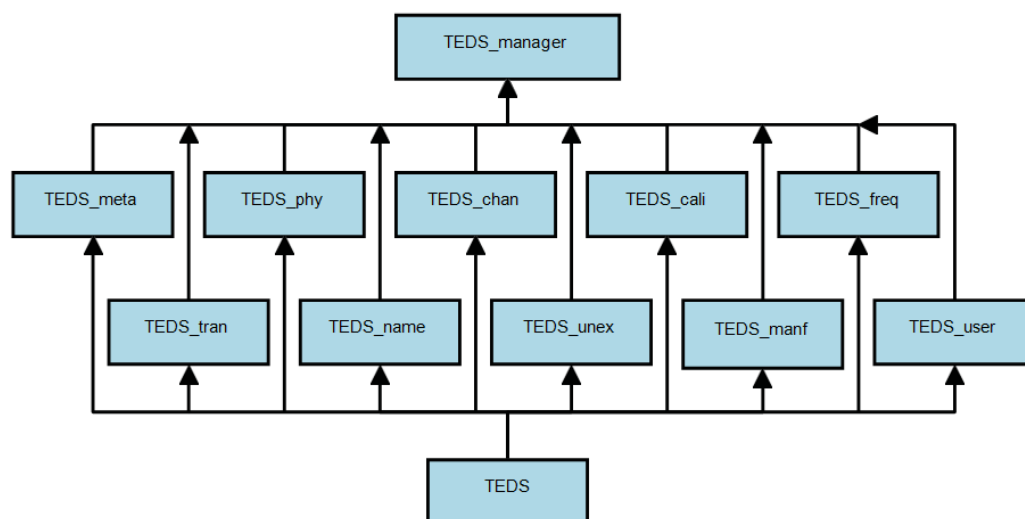


Figura 4.1 Estructura de datos TEDS.

4.1.2. Estructura de datos TEDS_manager

TEDS_manager, contiene punteros a las diferentes estructuras TEDS de ésta clase. Contiene bloques de memoria individuales y campo TEDS con tipo de dato *Boolean*, que indica los bloques de memoria TEDS existentes. Por otra parte, ésta clase incluye funciones para la gestión de estructuras TManager. Estas funciones son:

- TManagerInit.
- TManagerDelete.
- TManagerCreateMetaTEDS, TManagerCreatePhyTEDS, TManagerCreateChanTEDS, TManagerCreateCaliTEDS, TManagerCreateTranTEDS, TManagerCreateFreqTEDS, TManagerCreateNameTEDS, TManagerCreateUserTEDS, TManagerCreateUnexTEDS y TManagerCreateManfTEDS.

4.1.3. Intercambio de TEDS

Ésta biblioteca se utiliza para intercambiar datos entre módulos *NCAP* y *TIM*. Tiene una sola clase que tiene el mismo nombre. Esta clase no contiene ninguna estructura de datos, sólo las características que se requieren para el intercambio de datos. Hay tres funciones principales que se describirán en detalle y seis funciones auxiliares. Las tablas Tab. 4.2 y Tab. 4.3 indican estas funciones.

Tabla 4.2 Funciones principales.

Archivo	Función
exchangeCommandToTIM	Genera mandos al TIM
exchangeRespFromTIM	Reconoce la respuesta del TIM
exchangeTEDSToNCAP	Genera la respuesta

Tabla 4.3 Funciones auxiliares.

Archivo	Descripción
exchangeReadCommandClassFromNCAP	Reconoce los mandos de la clase
exchangeReadCommandFromNCAP	Reconoce el mando recibido
exchangeReadParam1FromNCAP	Lee los primeros parámetros del mando
exchangeReadParam2FromNCAP	Lee los segundos parámetros del mando
exchangeInit	Inicializa la máquina de estados
exchangeSendError	Genera una respuesta de error

Función *exchangeCommandToTIM*

En esta función se generan los mandos que se envían desde un módulo a otro *TIM* ó *NCAP*. La entrada a esta función es un puntero a un array de caracteres que se envía al módulo *TIM* y el tipo de memoria que desea accesar. La generación de órdenes se realiza mediante la máquina de estado. Esta máquina se divide en componentes de acuerdo con la memoria TEDS y si se quiere tomar un módulo *TIM*. Durante las llamadas repetidas, esta función, en un conjunto de caracteres, almacena los mandos para el módulo *TIM*. Los estados se almacenan en una variable global que puede ser inicializado usando la función llamada *exchangeInit*.

Función *exchangeRespFromTIM*

Esta función es declarada en el módulo de *NCAP* y el proceso en el módulo *TIM*. Para determinar la respuesta del *TIM*, se analiza el paquete enviado por el *NCAP*, canal de dirección del transductor, *acknowledge*, entre otros. Esto se logra con una máquina de estados. La entrada a esta función es un puntero al campo de datos recibida y el tipo de memoria que la recibe. Si el paquete del *NCAP* no le corresponde al *TIM* en cuestión, los datos no se guardan.

Función *exchangeTEDSToNCAP*

Esta función se declara en el módulo *TIM* para generar una respuesta a la orden de lectura desde el módulo de memoria TEDS *NCAP*. Tras la recepción de la orden, serán llamadas las funciones auxiliares correspondientes. Para luego, con una máquina de estado, determinar el tipo de orden, el tipo de memoria TEDS que se desea leer, y el conjunto de datos que desea enviar. Con este mecanismo, se generan los datos de la biblioteca TEDS de la respuesta, añade la cabecera y los envía el módulo *NCAP*. Si esta función no conoce el tipo de memoria TEDS, se genera un paquete de error y lo envía al módulo de *NCAP*. Las respuestas de error se ilustran en la Tab. 4.4

Tabla 4.4 Paquete de error.

Posición	Significado	Valor del octeto
0	Bandera Success/Fail	0
1	Longitud (más significativo)	0
2	Longitud (menos significativo)	0

4.1.4. Cómo utilizar la biblioteca

Antes de utilizar los *exchangeTEDS*, la biblioteca de módulos *NCAP* necesita inicializar la máquina de estado que genera los mandos para leer datos de un módulo. Ésto se logra con el uso de la función *exchangeInit*. Luego, se llama a la función *exchangeCommandToTIM* para que poder generar el mando al módulo *TIM*. Al recibir la respuesta del módulo *TIM*, se llama la función *exchangeRespFromTIM*, ésta función almacena los datos recibidos en la memoria TEDS. Se repite éste procedimiento hasta que la función *exchangeRespFromTIM* devuelve un valor *M_END*, lo que significa que ha tomado todos los datos de un área de memoria TEDS y la máquina de estado fue re-inicializada. Todos los posibles valores que la función devuelve, aparecen en la Tabla 4.5.

Tabla 4.5 Valores que regresa *exchangeRespFromTIM*.

Valor	Nombre del Bloque	Significado
-1	M_ERROR	Error en la respuesta
0	M_END	Límite de memoria excedido
1	M_OK	Respuesta correcta

Protocolo de comunicación IEEE 1451

La comunicación se logra mediante el envío de octetos agrupados en paquetes individuales. Están definidos diferentes tipos de paquetes de datos. *Command message* es usado para mandos de un módulo a otro y *Reply message* para la respuesta a los mandos. La transferencia de datos, desde los módulos de memoria TEDS entre *NCAP* y el módulo *TIM*, se lleva a cabo de modo que el módulo de *NCAP* envía una consulta ó un comando del módulo *TIM* y respuesta de *acknowledge*. Hay excepciones cuando el módulo *NCAP* envía un módulo de comando *TIM*, este comando se ejecuta, pero no responde a ella. Estos son, por ejemplo, mandos para ajustar el actuador, ajustes de controles internos ó mandos para cambiar el módulo *TIM* a otro estado. El protocolo de comunicación de datos entre los módulos *NCAP* y *TIM* se muestra en la Fig. 4.2.

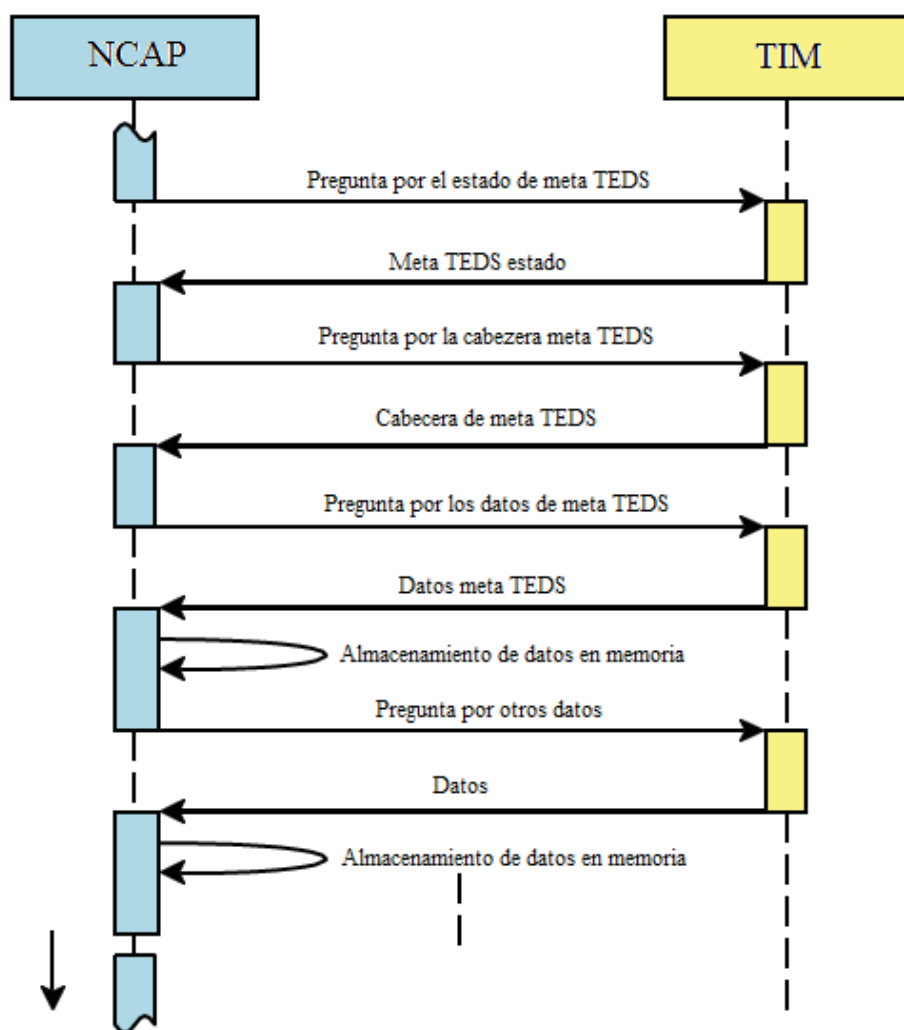


Figura 4.2 Protocolo de información entre los módulos *NCAP* y *TIM*.

4.1.5. Implementación

El trabajo Jan Popelka [24], implementa el IEEE 1451.2 para el protocolo CAN y su área de aplicación es la Aeroespacial, Fig. 4.3. El *TIM* lo implementa con un micro-controlador de 8 bits con los estándares IEEE 1451.0 e IEEE 1451.6. El *NCAP* lo implementa con una PC y los estándares IEEE 1451.0, IEEE 1451.6 e IEEE 1451.1. Aunque no sale a Internet.

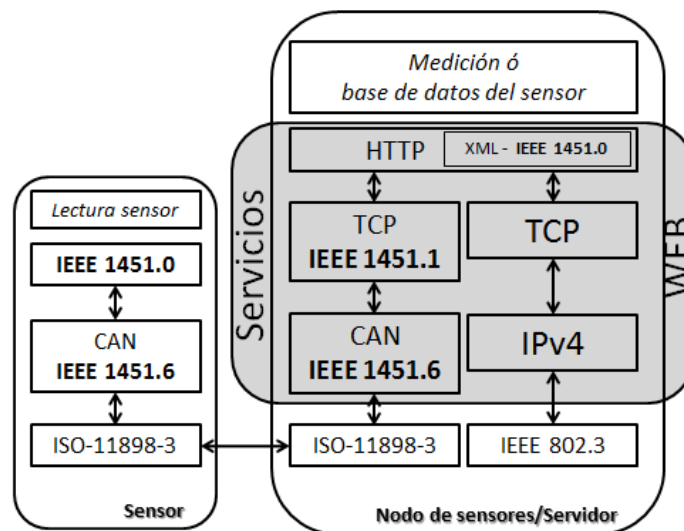


Figura 4.3 Interfaz IEEE 1451.x - Aeroespacial.

4.1.6. Conclusiones

La biblioteca del trabajo de tesis de Jan Popelka es un trabajo arduo, documentado y que se apega fielmente a lo que indica los manuales de la familia de estándares IEEE1451. Sin embargo en su implementación, lo hace exclusivamente para el protocolo *CAN* y la parte del *NCAP* lo hace en una PC [24]. Fue de gran ayuda encontrarlo, estudiarlo e implementarlo. Implementar el estándar IEEE 1451.5, una vez asimilado el estándar mismo y haber entendido las funciones de ésta biblioteca, fue relativamente fácil y no se requirió hacer ninguna modificación a la biblioteca.

Capítulo 5

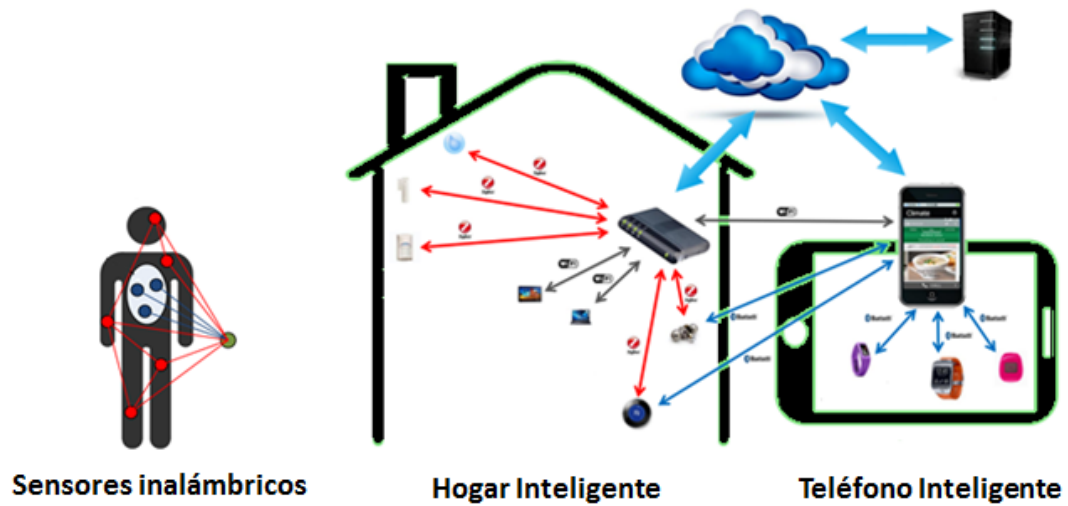
Implementación del prototipo *Gateway*

Para responder la pregunta de investigación descrita en el capítulo 1, se implementó el protocolo IEEE 1451 descritos en el capítulo 3, en los módulos de desarrollo 1322x-LPN (*Low Power Node*), 1322x-NCB (*Network Coordinator Board*) y el módulo tipo torre TWR-K60N512. Módulos ofrecidos por la compañía *Freescale*. Para fines de ilustración se muestra nuevamente la Figura 1.3, pero con la ubicación de los protocolos y módulos en la Fig. 5.1.

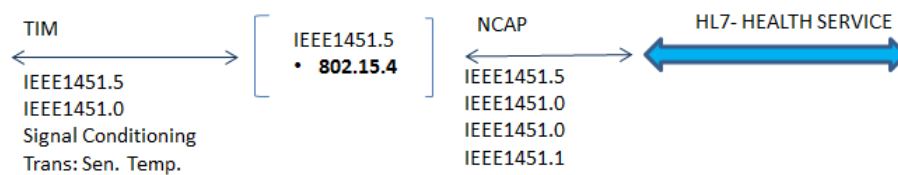
5.1. Descripción del *Hardware*

Freescale ha desarrollado la más completa plataforma de desarrollo para la evaluación del micro-controlador MC1322X, el cual es una solución de *PiP* plataforma contenida en el empaque (*PiP* - por sus siglas en inglés). Permite al desarrollador seleccionar el *Hardware* y *Software* que mejor cumpla con sus necesidades; este kit le permite hacer la evaluación casi de manera inmediata. El *PiP* lo integran un transceptor de 2.4 GHz, un núcleo ARM7TDMI, memoria, puertos de entrada/salida y componentes de radio frecuencia, todo esto en un sólo empaque. Esto hace que se reduzcan significativamente los costos y el tamaño. El *Kit* de desarrollo MC1322X es la plataforma ideal para aplicaciones más complejas con protocolos IEEE 802.15.4 y ZigBee, otorgando mucha capacidad y expansión de memoria. El *Kit* MC1322X lo integran los siguientes módulos:

1. 1322X-SRB (*Sensor Reference Board*) - Contiene un circuito integrado (*CI* - por sus siglas en inglés) MC13224 *PiP*, acelerómetro MMA7260Q de tres ejes, sensor de presión MPXV5010G y un sensor de temperatura.
2. 1322X-NCB (*Network Coordinator Board*) - Contiene un circuito integrado (*CI* - por sus siglas en inglés) y una pantalla LCD (*Liquid Crystal Display* - por sus siglas en



(a) Hogar y sensores inteligentes.



(b) Familia IEEE 1451.

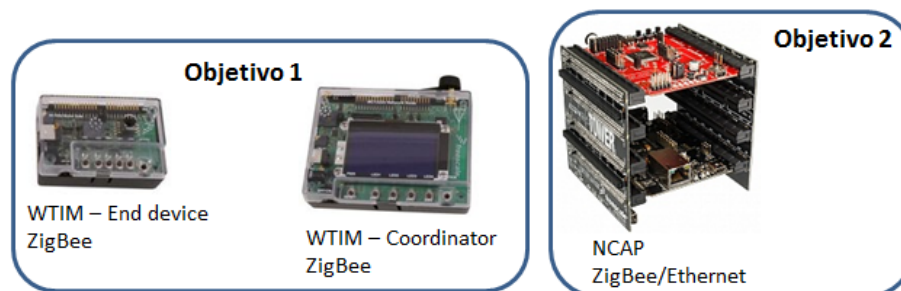
(c) *Kits* de desarrollo Freescale.

Figura 5.1 Implementación propuesta.

inglés) para monitoreo de mensajes. Es una plataforma ideal para el desarrollo de un coordinador de red.

3. 1322X-LPB (*Low Power Board*) - Contiene un *CI* MC13224 y es ideal para mediciones de consumo de potencia. Es posible operarlo con baterías de tamaño AAA ó con una batería tipo moneda, usa y tiene la opción de un convertidor de energía tipo *buck* para la opción de bajo consumo de energía.
4. 1322X-USB (*Universal Serial Bus*) - Este módulo está programado para que funciones como un *sniffer* de paquetes con protocolo 802.15.4/ZigBee. Puede ser re-programado para otras aplicaciones.

Para fines de comprobación, sólo se emplearon los módulos 1322X-LPB y 1322X-NCB, los cuales se describen a continuación.

5.1.1. Módulo MC1322x-LPB

El MC1322x-LPB es un módulo de sensor inalámbrico que cumple con el estándar IEEE 802.15.4, su núcleo es un dispositivo *PiP* MC1322x de *Freescale* puede ser usado para conectividad punto a punto o de malla. El MC1322x está diseñado para soluciones totales, alta capacidad de procesamiento y bajo consumo de potencia. Cuenta con conector de entradas y salidas de propósito general (*GPIO* - por sus siglas en inglés), un segundo conector para comunicación serie (*UART* - por sus siglas en inglés) y un puerto *JTAG* (por sus siglas en inglés) para depuración del programa. El módulo se muestra en la Fig. 5.2.

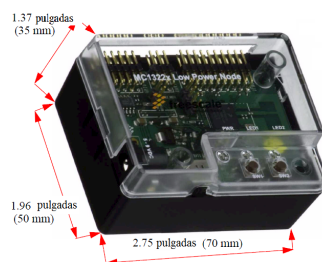


Figura 5.2 Nodo de bajo consumo de potencia MC1322x-LPB.

5.1.2. Módulo MC1322X-NCB

El MC1322x-NCB es un módulo de sensor inalámbrico que cumple con el estándar IEEE 802.15.4, su núcleo es un dispositivo *PiP* MC1322x de *Freescale*, puede ser usado

para conectividad punto a punto o de malla. El MC1322x está diseñado para soluciones totales, alta capacidad de procesamiento y bajo consumo de potencia. Cuenta con conector de entradas y salidas de propósito general *GPIO*, un segundo conector para comunicación serie *UART*, un puerto *JTAG* para depuración del programa, con conexión a computadoras personales (*PC* - por sus siglas en inglés) u otros dispositivos. Se diferencia del MC1322x-LPN, pero con mayor capacidad de memoria y periféricos con lo que puede fungir como nodo coordinador. El módulo se muestra en la Fig. 5.3.

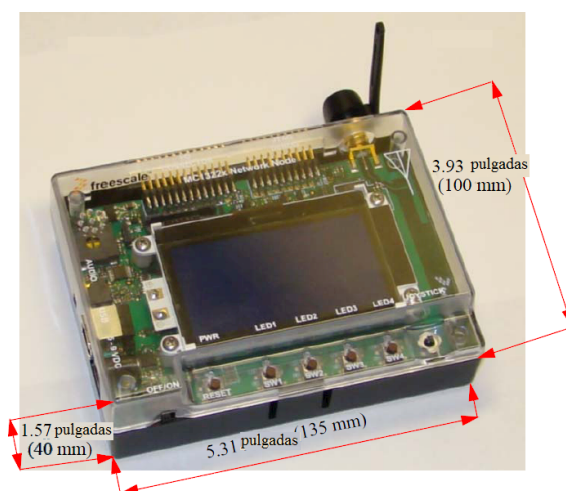


Figura 5.3 Nodo de red MC1322x.

5.1.3. Sistema de torre TWR-K60

El módulo *TWR-K60N512* es un módulo de control compatible con el sistema de torre de *Freescale*. Cuenta con un micro-controlador de arquitectura ARM® Cortex™-M4, con un dispositivo USB 2.0 y con un módulo de Ethernet 10/100 Mbps. La Fig. 5.4 muestra la imagen de éste módulo.

Características

La siguiente lista resume las características del módulo:

- Compatible con el sistema de torre.
- Micro-controlador MK60N512VMD100: K60N512 de 144 pines a 100MHz de operación.
- Circuito de depuración *JTAG* con puerto serie virtual.
- Acelerómetro de tres ejes *MMA7660*.

- Cuatro (4) LEDs programables por el usuario.
- Cuatro (4) zonas táctiles de tipo capacitivo.
- Potenciómetro.
- Ranura para memoria flash tipo *SD*.

El micro-controlador MK60N512VMD100 cuenta con las siguientes características.

- Arquitectura ARM Cortex-M4 de 32-bit con instrucciones *DSP*.
- Frecuencia máxima de operación de 100MHz.
- 144 pines tipo MAPBGA, dimensiones de 13mm x 13mm.
- Rango de operación de voltaje de entrada 1.71V – 3.6V.
- 512 Kbytes de memoria programable tipo *flash*, 128 Kbytes memoria estática tipo *RAM*.
- Interfaz de bus externo.
- Convertidor analógico digital de 16-bit con *SAR*, convertidor digital analógico de 12-bits.
- Comparador analógico de alta velocidad con convertidor digital analógico de 6-bit.
- Voltaje de referencia programable.
- Ethernet 10/100 Mbps.
- SPI, I2C, UART, CAN, GPIO, DMA.
- Interfaz de depuración: JTAG, cJTAG, SWD.

5.2. Descripción herramientas de *Software*

5.2.1. Conectividad Inalámbrica con BeeKit

La herramienta de desarrollo para la conectividad inalámbrica de *Freescale Beekit* es una aplicación de *Software* diseñada para crear, modificar, guardar y actualizar soluciones de redes inalámbricas. *Beekit* provee una interfaz tipo *wizard* para configurar parámetros fácilmente antes de generar el proyecto. Contiene bibliotecas para la conectividad inalámbrica y plantillas de aplicación/ejemplos. El usuario podrá importar esta configuración a herramientas de desarrollo de programación (*IDE* - por sus siglas en inglés) diseñadas en el módulo *codebase* requerido por el *Beekit*. *Beekit* otorga una licencia sin límite para *SMAC*, IEEE 802.15.4 PHY/MAC y BeeStack™ ZigBee. Los módulos *codebases* son aplicaciones de programación embebida y no se describen con detalle en este trabajo. La Fig. 5.5, muestra la pantalla principal.



Figura 5.4 Sistema torre (TWR) de *Freescale*.

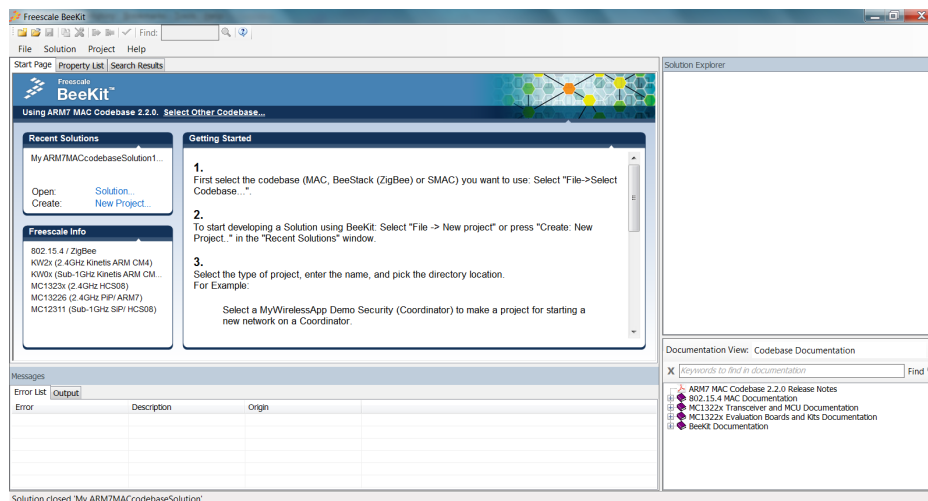


Figura 5.5 Ventana principal de *BeeKit*.

Características

- Validación automática de configuración de parámetros del proyecto.
- Generación de archivos de trabajo que pueden ser importados en un ambiente integrado de programación *IDE*.
- Escalable, soporta nuevos *codebases* y funcionalidades
- Cuenta con aplicaciones para: SMAC, IEEE 802.15.4 MAC, Freescale BeeStack, RF4CE, SynkroRF.
- Soporta micro-controladores HCS08 y ARM7.
- Documentación disponible para todo usuario.

5.2.2. IAR Systems

La interfaz integrada de ambiente de desarrollo IAR Systems®, es actualmente el líder mundial de herramientas de desarrollo para sistemas dedicados o embebidos. El compilador para C/C++ y herramientas de depuración *IAR Embedded Workbench* está disponible para micro-controladores de 8, 16 y 32 bits; incluyendo toda la gama de arquitectura ARM. La Figura 5.6 ilustra los paquetes de herramienta que incluye.

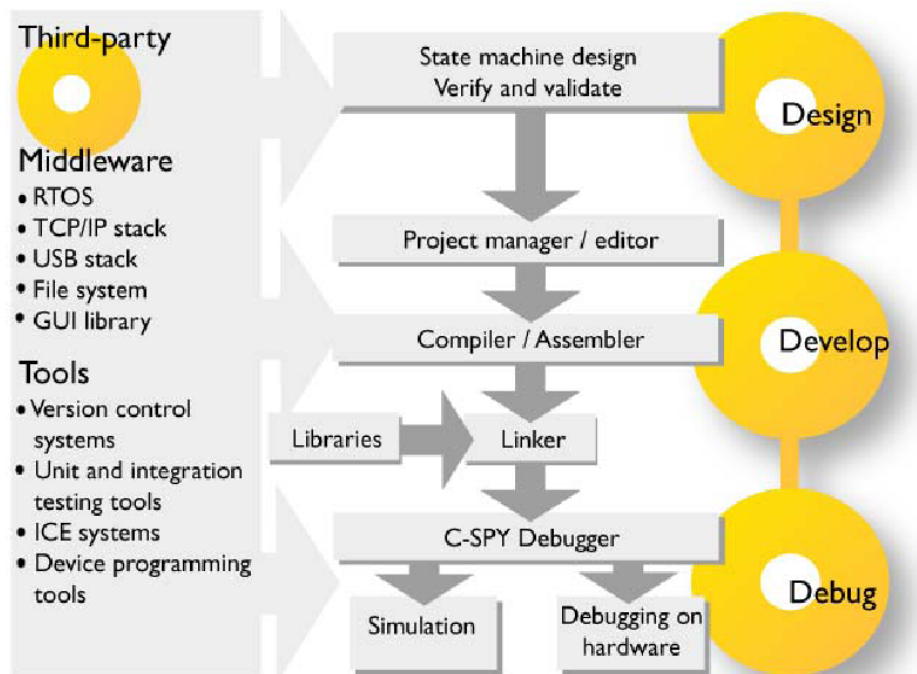


Figura 5.6 Herramientas integradas en IAR Systems, tomada de [14].

IAR Embedded Workbench

IAR Embedded Workbench otorga un ambiente de desarrollo integral que le permite al programador administrar un proyecto de aplicación para sistemas dedicados. Este *IDE* contiene todas las herramientas necesarias para el proyecto de aplicación: compilador C/C++, ensamblador, enlazador, editor, administrador de proyectos y depurador del programa IAR C-SPY®. La Figura 5.7 muestra la ventana de inicio.

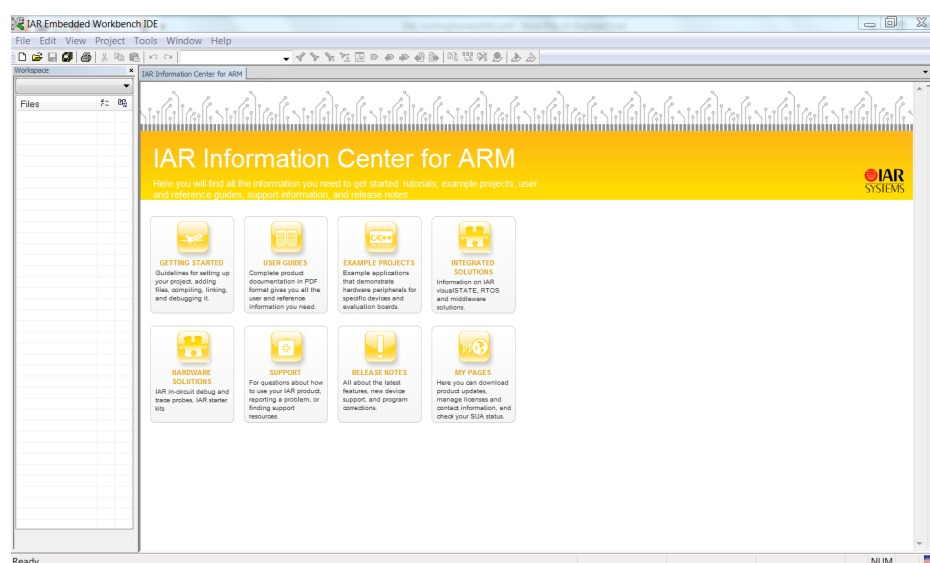


Figura 5.7 Ventana de inicio de IAR Systems.

IAR visualSTATE

IAR visualSTATE integra un conjunto de herramientas, basadas en máquinas de estado, para diseño, prueba e implementación de aplicaciones dedicadas. Provee herramientas formales de verificación y validación. También es capaz de trasladar éste trabajo a código de lenguaje de programación C.

I-jet

I-jet es una herramienta de depuración por circuitería, también conocida como depurador *JTAG*, para ARM7/ARM9. Se conecta a la PC por medio de un puerto USB y al microcontrolador en cuestión, permitiendo depurar programas dedicados. El cual está integrado en el sistema *IAR Embedded Workbench*. IAR j-link otorga un buffer dedicado para rastreo (*ETB* - por sus siglas en inglés), lo que permite rastrear instrucciones del programa y disponible para el depurador a través del puerto *JTAG*. El depurador se muestra en la Fig. 5.8.



Figura 5.8 Programador j-link.



Figura 5.9 Teraterm logotipo y configuración del puerto serie.

5.2.3. Teraterm

Teraterm es una aplicación de emulador de terminal con código abierto, sin costo. Emula diferentes tipos de terminales, desde la DEC VT100 hasta la DEC VT382. Soporta telnet, SSH 1 y 2, y una conexión a puerto serie. La Fig. 5.9 muestra el logotipo y la ventana de configuración del puerto serie.

5.3. Aplicación *MyStarNetwork Demo*

Esta aplicación, de fácil implementación con BeeKit denominada *MyStarNetwork Demo*, consiste en la concentración de tres proyectos individuales: *MyStarNetwork Demo* (Coordinador), *MyStarNetwork Demo* (End Device Sensor Node) y *MyStarNetwork Demo* (End Device Low Power Node). Está compuesto de las siguientes partes: *MacStandalone*, *MacApps*, *MyStarNetwork Demo*, *StarNetworkDemoCoordinator* y *StarNetworkDemoEnd-Device*. Los tres primeros son comunes en los tres proyectos y el último es igual para los

proyectos End Device Sensor Node y End Device Low Power Node. La ilustración gráfica se muestra en la Figura 5.10.

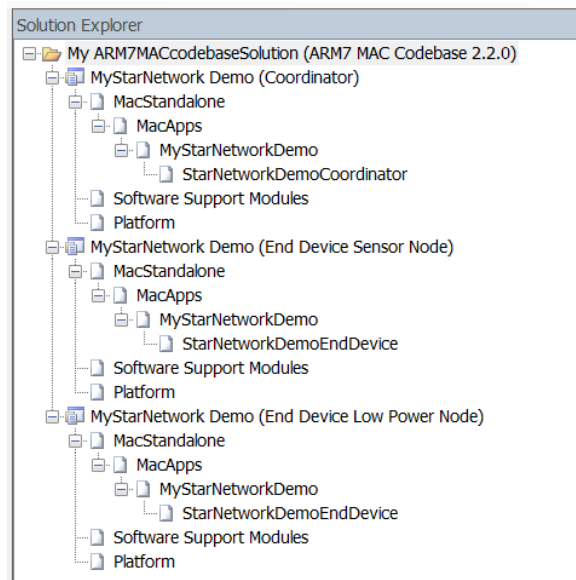


Figura 5.10 Archivos del proyecto MyStarNetworkDemo.

5.3.1. MacStandalone

Se refiere al componente 802.15.4 MAC/PHY, el cual es la implementación de la capa física del estándar IEEE 802.15.4 y la sub-capa de control de acceso al medio MAC. Se usa para transmisión inalámbrica de baja transferencia para dispositivos fijos, portátiles y móviles sin batería ó consumo de energía restringido, operando típicamente en el rango de espacio personal (POS). Este componente es proporcionado por BeeKit como biblioteca sin código fuente. MAC 2003.

5.3.2. MacApps

Este componente está integrado como *template* en el *CodeBase*. Se entrega el código fuente y puede ser usado como ejemplo. Con ésto, el usuario fácilmente puede crear aplicaciones arriba de la capa MAC. Es importante señalar, que el código proporcionado por MacApps es lo más independiente posible del *hardware* y *MCU*. Por lo que es posible crear aplicaciones para otros componentes con pocos cambios. Canal: 0x07FFF800 Dirección Extendida: 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF.

5.3.3. MyStarNetwork Demo

Esta demostración implementa una red con el protocolo IEEE 802.15.4. El sistema está compuesto por un coordinador *PAN* y uno ó más *End Devices* hasta un límite máximo de 4. Una vez programados, sólo hay que energizarlos. El intercambio de datos se dará inmediatamente después de que el primer *End Device* se asocie al coordinador. La red se extenderá dinámicamente al irse asociando otros *End Devices*. Cuando el coordinador recibe los datos, éstos son guardados en *buffers* para cada uno de los *End Devices* asociados. El límite de buffers en la MAC es cuatro; de ahí el límite de *End Devices* también de cuatro. Una vez llenos los *buffers*, los paquetes serán desechados, por lo que la información de éstos paquetes se perderán. El programa manda un mensaje de alerta en la consola del coordinador. La interacción de ésta aplicación con el usuario, se dará a través de mensajes en la consola del coordinador y el puerto serie del *End Device*. Los mensajes que aparecen en ésta consola son transmitidos a todos los *End Devices* asociados. El mensaje es una serie de caracteres, limitados por el caracter retorno de carro, hasta un máximo de 15. Debido a la función de bajo consumo de energía habilitada del *End Device* al coordinador, sólo un mensaje predefinido es enviado cuando el usuario presione un interruptor de la tarjeta del *End Device*. Los usuarios podrán monitorear los mensajes de ésta aplicación a través de una terminal ó un dispositivo *sniffer* que esté en el canal de ésta red. Además de transmitir regularmente información del *End Device* al coordinador, éste envía un valor tiempo cada tres segundos a todos los dispositivos.

5.3.4. MyStarNetworkDemoCoordinator

Ésta es la implementación del coordinador de la red de área personal de *My Star Network Demo*. Este ejemplo demuestra la transferencia indirecta de datos del coordinador al *end device*. Los datos enviados al *end device* no son transmitidos inmediatamente, son puestos en la cola de la MAC del coordinador. Y aquí permanecerá hasta que el *end device* haga solicitud. Esta solicitud es iniciada por la red ó la capa de aplicación del *end device*. Dirección corta: 0xCAFE PAN ID: 0xBEEF

5.3.5. MyStarNetworkDemoEndDevice

Ésta es la implementación del *end device* de *My Star Network Demo*. Este ejemplo demuestra la transferencia de datos entre el *end device* y el coordinador de *PAN*. En el estándar IEEE 82.15.4 la mayor parte de la comunicación en la red es llevada a cabo por el

end device. Los *end devices* son normalmente energizados por baterías, por lo que necesitan controlar el flujo de datos para prolongar la vida de la batería. Esto es llevado a cabo enviando los datos directamente al coordinador, este por su parte la solicita por medio de *polling*. El coordinador envía los datos a un *end device*, sólomente si sabe que el *end device* está escuchando (*listening*), *id.est.* cuando el *end device* ha solicitado los datos. Para comprobar la transferencia de datos del coordinador al *end device*, ambos deberán estar conectados a una PC con terminal RS232. El puerto serie deberá estar configurado con 8 bits de datos, 1 bit de parada y sin paridad. La razón de transferencia se deja al usuario. Al enviar un archivo en código ASCII de la terminal al coordinador, el archivo aparecerá en la terminal conectada al *end device*.

Los pasos requeridos para recibir el paquete de datos del coordinador son:

1. Asociarse al coordinador del que queremos recibir los datos.
2. Generar una petición de mensaje tipo MLME-Poll y poner en el coordinador la información obtenida durante el *Active Scan*. Enviar el mensaje al MLME.
3. Si se recibe una confirmación de *poll* diferente de exitosa, apaga el *transceiver* durante un período corto de tiempo. De lo contrario, se manda el mensaje, después de eso se apaga el *transceiver* y el *MCU* en cierto período de tiempo.
4. Despertar y empezar nuevamente desde el punto número 2.

5.3.6. Exportando el proyecto

Una vez creado el proyecto MyStarNetwork (Coordinator) y MyStarNetwork (End Device) con el paquete BeeKit de Freescale, se exporta la solución al entorno de desarrollo IAR Embedded Workbench. BeeKit muestra la opciones de exportación y crea un proyecto con configuración de archivos para ser leídos y compilados en el IDE. Para ello se realiza lo siguiente:

1. De la barra del menú de opciones de la ventana principal de BeeKit Figura 5.5, seleccionar Solution->Export Solution...
Apareciendo la siguiente ventana de la Fig. 5.11.

2. Asegurarse que el proyecto que se exportará esté seleccionado y presionar el botón *OK*. Durante el proceso de exportación, varios mensajes se desplegarán en una parte de la ventana principal de *BeeKit*.

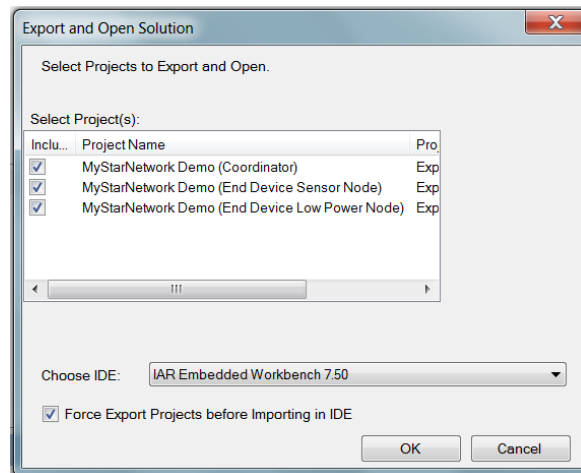


Figura 5.11 Ventana para exporta el proyecto BeeKit.

5.4. IEEE 1451 en IAR

Para la implementación del estándar IEEE 1451 en los módulos *TIM* y *NCAP*, se importa el proyecto generado por BeeKit y se agregan al proyecto los archivos de la librería Jan Popelka para cada uno de los módulos. Esto se explica a detalle en las siguientes secciones. También se explican las complicaciones encontradas y la manera que se solucionaron.

5.4.1. Importar el proyecto BeeKit en IAR

Para importar un proyecto generado por BeeKit en IAR Embedded Workbench se siguen los siguientes pasos:

1. Del menú de la ventana principal de IAR, seleccione: File->Open->Workspace.
2. En la ventana de Open Workspace, busque la solución que fué exportada y seleccione el archivo exportado (terminación .eww).
3. El proyecto se cargará al ambiente de desarrollo de IAR EWARM, como lo muestra la Figura 5.12.

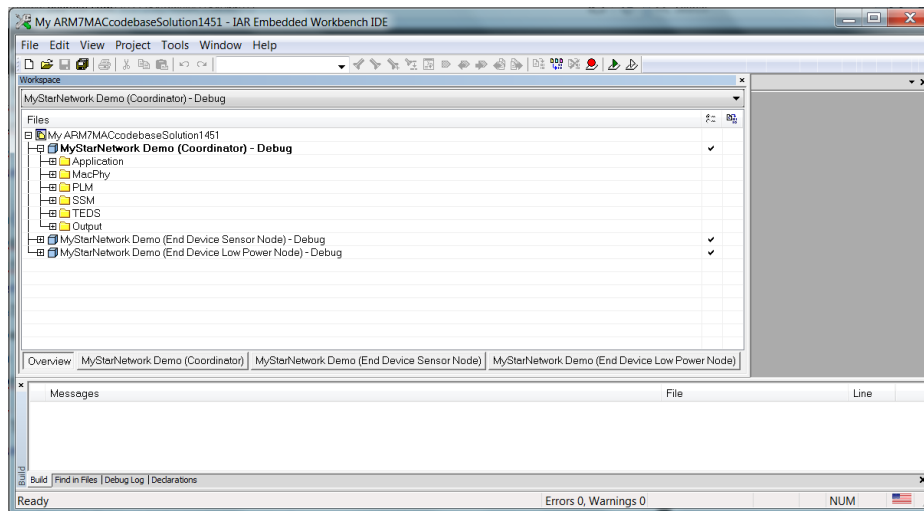


Figura 5.12 Ventana IAR con proyecto importado de BeeKit.

5.4.2. Agregar los archivos al proyecto

Se agregaron los archivos de la biblioteca IEEE 1451 y compilaron hasta que lo hiciera sin errores. Esto para cada uno de los proyectos *id. est.* MyStarNetwork Demo (Coordinator), MyStarNetwork Demo (End Device Sensor Node) y MyStarNetwork Demo (End Device Low Power Node), como aparece en la Figura 5.13.

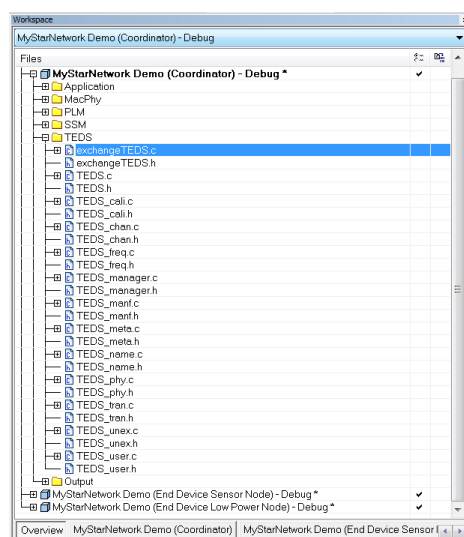


Figura 5.13 Ventana IAR con IEEE 1451 en el proyecto.

5.4.3. Configuración TIM y NCAP

El trabajo realizado por Jan Popelka, nos permite configurar los módulo MC1322x-LPB y el módulo MC1322x-NCB como *TIM* ó *NCAP* con sólo habilitar la línea no. 34 del archivo exchange TEDS del módulo correspondiente. La Figura 5.14 ilustra este paso.

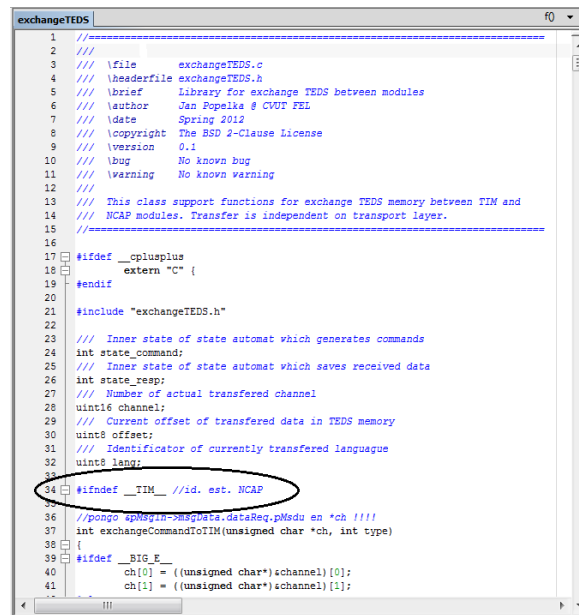


Figura 5.14 Configuración *TIM* ó *NCAP*.

5.4.4. Inicialización de TEDS

La declaración de los TEDS, así como la inicialización se lleva a cabo en el archivo MApp.c, función **void MApp_init(void)**. A continuación, sólo se explicará el caso de meta TEDS, el mecanismo es igual para los demás excepto, claro está, para la información del TEDS en cuestión.

En la línea número 238 de la Fig. 5.15, el apuntador va al archivo TEDS_meta.c y accesa la función **TEDSMetaInit()**. En donde principalmente: las variables toman valores iniciales, se genera la longitud y *checksum* del meta TEDS (líneas 464 y 465 respectivamente) de la Fig. 5.16

5.4.5. Envío de TEDS - TIM

Para enviar la información de los TEDS del *TIM* al *NCAP*, se realiza en dos partes:

```

216 static TEDS_meta mymeta; //Variable global.
224 void MApp_init(void)
225 {
226     TEDS_meta* p_mymeta;
232     p_mymeta = &mymeta;
238     TEDSMetaInit(p_mymeta);
    ...
285 }

```

Figura 5.15 Declaración y manejo de TEDS.

```

415 void TEDSMetaInit(TEDS_meta* tm)
416 {
432     tm->Length = 0x0;
433     tm->Id.Family = 0x00;
434     tm->Id.Class = 0x00;
435     tm->Id.Version = 0x00;
436     tm->Id.Tuple_length = 0x00;
    ...
464     TEDSMetaGenerateLength(tm);
465     TEDSMetaGenerateChecksum(tm);
466 }

```

Figura 5.16 Inicialización del Meta TEDS.

1. Usar las funciones que ofrece el archivo *exchangeTEDS.c* correspondiente a la información del TEDS que se desea transmitir.
2. Colocar esa información en el *payload* del protocolo 802.15.4. En el caso de nuestra aplicación, ésto se lleva a cabo en la variable *mpPacket->msgData.dataReq.pMsdu*.

Para la comprobación del envío de datos, se modificó el código de la función *App_SendSensorStatus()*, ver Fig. 5.17, sólo se muestran las líneas de código que corroboran los puntos 1 y 2 mencionados en el párrafo anterior.

La parte más importante, referente al estándar IEEE 1451, es la línea número 1186, donde la función *exchangeTEDSToNCAP()*, se encarga de dos cosas:

1. Poner la información que queremos transmitir en la variable *pMsdu*. En este caso, lectura del ID del meta TED.

```
1161 static void App_SendSensorStatus(void)
1162 {
1163     int mi_envio;
1164     mpPacket = NULL;
1165     ...
1175     if(mpPacket != NULL)
1176     ...
1186     mi_envio = exchangeTEDSToNCAP(mpPacket->msgData.dataReq.pMsdu,
1187                                   TEDS_META, COMAND_READTEDS, TM_TEDSID);
1195     mpPacket->msgData.dataReq.msduLength = mi_envio;
1196     ...
1207     (void) MSG_Send(NWK_MCPS, mpPacket);
1208     ...
1212 }
```

Figura 5.17 Transmisión del Meta TEDS.

2. Regresa el valor de número de bytes, variable *mi_envio*, que tiene esta información. Este valor es requerido por el estándar IEEE 802.15.4 y se le proporciona en la variable *msduLength* indicado en la línea número 1207.

Finalmente se le entrega al estándar 802.15.4, línea 1207, para que genere las condiciones para su transmisión.

5.4.6. Función *exchangeTEDSToNCAP()*

En esta sección se explicarán solamente las líneas de código relacionadas con la Fig. 5.17. La tarea de ésta función es llenar la cadena de caracteres con la información de los TEDS y regresar el número de *bytes* que se requieren enviar. También cuenta con mecanismos y una máquina de estados para saber el tipo de TEDS y, si se requiere, crear y/o inicializar, entre otras cosas. Parte del listado se presenta en la Fig. 5.18, al final se analizan los puntos importantes.

Para este ejemplo, el llenado queda de la siguiente manera:

- `ch[0] = channel[0];` - primer *byte* de dirección indicada en el TEDS channel. Línea 42.
- `ch[1] = channel[1];` - segundo *byte* de dirección indicada en el TEDS channel. Línea 43.
- `ch[2] = 0x1;` - así lo marca el estándar. Línea 48.

```
23 extern TEDS_meta mymeta;
39 int exchangeCommandToTIM(unsigned char *ch,int type)
40 {
42   ch[0] = ((unsigned char*)&channel)[0];
43   ch[1] = ((unsigned char*)&channel)[1];
48   ch[2] = 0x1;
49   ch[6] = type;
50   switch(type)
51   {
52     case TEDS_META:
53     {
54       switch(state_command) {
56         case 0:{
58           ch[3] = 0x1;
59           ch[4] = 0x0;
60           ch[5] = 0x1;
61           state_command++;
62           TManagerCreateMetaTEDS(&manager);
63           return 7; } //case 0
65         case 1: { ... }
74         case 2: { ... }
83         case 3: { ... }
92         case 4: { ... }
101        case 5: { ... }
110        case 6: { ... }
119        case 7: { ... }
128        case 8: { ... }
137        case 9: { ... }
146        case 10: { ... } } } //case TEDS_META
157        case TEDS_CHAN: { ... }
450        case TEDS_UNEX: { ... }
519        case TEDS_NAME: { ... }
599        case TEDS_USER: { ... }
631        case TEDS_FREQ: { ... }
690        case TEDS_MANF: { ... }
714        case TEDS_PHY: { ... }
738        case TEDS_CALI: { ... }
842        case TEDS_TRAN: { ... } } } // switch(type)
974   return 0;
975 }
```

Figura 5.18 Cadena del Meta TEDS.

- `ch[3] = 0x1`; - así lo marca el estándar. Línea 58.
- `ch[4] = 0x0`; - así lo marca el estándar. Línea 59.
- `ch[5] = 0x1`; - así lo marca el estándar. Línea 60.
- `ch[6] = 0x1`; - valor de `TEDS_META` asignado por el estándar; este dato lo proporciona quien llama a ésta función. Línea 49.
- `return 7`; - finalmente regresa el valor 7, que son los datos que se llenaron en la variable `ch[]`.

El listado también muestra las opciones de llenado para otros TEDS, son accedidos con el segundo término de ésta función *type*.

5.4.7. Recepción de TEDS - NCAP

La recepción de éstos datos por el *NCAP*, se lleva a cabo en el archivo *MApp.c* con la función `static void App_HandleMcpsInput(mcpsToNwkMessage_t *pMsgIn)`. El cual se muestra en la Fig. 5.19, sólo con información para la explicación de su funcionamiento.

```

902 static void App_HandleMcpsInput(mcpsToNwkMessage_t *pMsgIn)
903 {
904     int TEDSID_ok;
906     switch(pMsgIn->msgType) {
953         case gMcpsDataInd_c: {
973             TEDSID_ok =
                TEDSSetTedsId((uint8_t*)&pMsgIn->msgData.dataInd.pMsdu[0], &mymeta.
                    Id);
981         break; } //case gMcpsDataInd_c
989     }
990 }
```

Figura 5.19 Recepción del Meta TEDS.

En la función `App_HandleMcpsInput()`, línea 902, se guarda la información que envió el *TIM* al *NCAP*, a través de la variable `pMsgIn`. La tarea de la función `TEDSSetTedsId()`, línea 973, es tomar los datos de la variable `pMsdu` y guárdalos en la variable `mymeta.Id`. También cuenta con un mecanismo de verificación para saber si fueron grabados exitosamente. En la Fig. 5.20, se muestra parte de lo señalado. Es importante señalar que esta función regresará *FALSE*, línea 32, si el dato `ch[x]` no es `0x03`, *id. est.* no es el identificador de TEDS ID. El resto del código llena la estructura `mymeta` con los datos que le envió el *TIM*.

```
27 boolean TEDSSetTedsId(unsigned char* ch, TEDSID *id)
28 {
29     int i, length;
30     uint8* pui, *pch;
31     pui = (uint8*)id;
32     if(!(*pch++==0x03))
33         return FALSE;
34     length = (int)*pch++;
35     for(i = 0; i < length; i++)
36         *pui++ = *pch++;
37     return TRUE;
38 }
```

Figura 5.20 Función TEDSSetTedsId.

5.5. Resumen

Este capítulo describe los elementos físicos y las herramientas de desarrollo con las que implementó en estándar IEEE 1451.0, IEEE 1451.1 y IEEE 1451.5. Beekit y BJP fueron elementos clave que redujeron significativamente el tiempo para su implementación en los módulos TIM y NCAP.

Capítulo 6

Resultados

En éste capítulo se hace una descripción de la implementación de TEDS requeridos para la lectura de un sensor de temperatura, la cual es realizada con un módulo *TIM* y un módulo *NCAP*. Se muestra como se manda la información enviada por el *TIM* al *NCAP* y cómo éste la recibe y procesa. Con ello, se considera la parte esencial de la arquitectura de e-Health propuesta.

Los resultados mostrados en éste capítulo corroboran la parte correspondiente a los WSN de la arquitectura propuesta en el capítulo 1.

6.1. TEDS

A continuación se describe la estructura de los TEDS para un *TIM*. En el *TIM* se implementa un canal (channel) para la información de la temperatura de un termistor. Para la calibración se implementa un regresión lineal dentro del rango de operación. El envío de los TEDS se hacen cada 3 segundos.

La sección 5.5 de [1] especifica que todos los TIMs deberán implementar los siguientes TEDS: Meta-TEDS, TransducerChannel TEDS, User's Transducer Name TEDS y PHY TEDS. El formato de PHY TEDS está fuera del alcance de éste estándar, por lo que no se describirá en este trabajo.

6.1.1. Meta-TEDS

La Tab. 6.1 muestra la información que se implementó para Meta-TEDS. La descripción detallada de cada uno de los campos se encuentra en el apéndice O de [1].

Tabla 6.1 Meta-TEDS

Descripción del campo	Bytes
Longitud total	00 00 00 24
Encabezado IDentificador	03 04 00 01 01 01
UUID	04 0A 81 C0 F9 74 48 81 F5 62 2E 78
Tiempo fuera de operación	0A 04 3F 00 00 00
Tiempo de auto-prueba	0C 04 C0 A0 00 00
Número de canales	0D 02 00 01
Checksum	F8 82

Una manera de introducir la información de los TEDS es directamente en el código del programa. En la Fig. 6.1 se muestra parte del código para introducir los datos del campo UUID del Meta-TEDS.

```

438 void MApp_init(void)
439 {
440     TEDS_meta* p_mymeta;
447     p_mymeta = &mymeta;
453     TEDSMetaInit(p_mymeta);
462     p_mymeta->Uuid[0] = 0x81;
462     p_mymeta->Uuid[1] = 0xc0;
462     p_mymeta->Uuid[2] = 0xf9;
462     p_mymeta->Uuid[3] = 0x74;
462     p_mymeta->Uuid[4] = 0x48;
462     p_mymeta->Uuid[5] = 0x81;
462     p_mymeta->Uuid[6] = 0xf5;
462     p_mymeta->Uuid[7] = 0x62;
462     p_mymeta->Uuid[8] = 0x2e;
462     p_mymeta->Uuid[9] = 0x78;
524 }
```

Figura 6.1 Datos del parámetro UUID.

6.2. Envío al NCAP

La Fig. 6.2 muestra el código y el mapa de memoria para el registro de los datos del metaTEDS ID, que serán enviados al NCAP. En la parte superior aparece el código de la función *TEDSGetTedsID()*, en la parte inferior el mapa de memoria que son cargados con éstos datos.

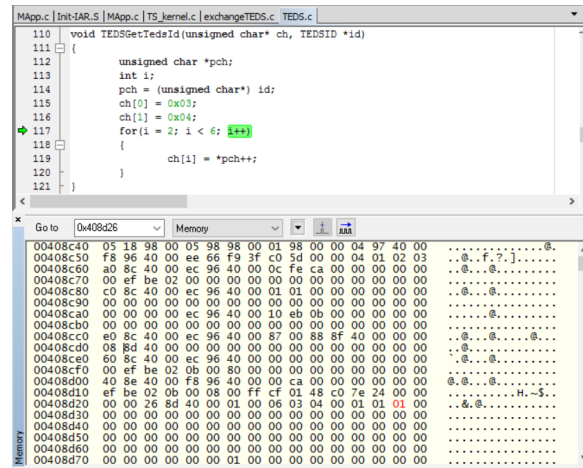


Figura 6.2 Meta_TEDS TIM al NCAP.

Para el envío de ésta información al *NCAP*, se hace con la función *exchangeTEDSToNCAP()*. Esta función le indica a la BJP que cargue los datos del ID del META-TEDS con el mando de lectura de TEDS, los cuales son puestos en la variable *pMsdu*, como se muestra en la línea 1186 de la Fig. 6.3. La función regresa un entero con el valor del número de caracteres que se enviarán, el cual depende de la información solicitada. Este dato es requerido por el estándar IEEE 802.15.4 y se proporciona a través de la variable *msduLength*, línea 1195. El envío de ésta información se lleva a cabo por la función *MSG_Send(NWM_MCPS, mPacket)*, línea 1207, la cual cumple con el estándar IEEE 802.15.4.

6.3. Recepción del NCAP

La Fig. 6.4 muestra los datos recibidos por el *NCAP*, los cuales corresponden a los enviados por el *TIM*.

La manera de obtener la información enviada por el *TIM*, es a través de la función *App_HandleMcpsInput()* específicamente la variable *pMsdu*. Para pasar ésta información al metaTEDS del *NCAP*, se hace uso de la función *TEDSSetTedsId()*, donde tomará los datos

```

1161 static void App_SendSensorStatus(void)
1161 {
1186  mi_envio = exchangeTEDSToNCAP(mpPacket->msgData.dataReq.pMsdu,
                                TEDS_META, COMAND_READTEDS, TM_TEDSID);
1195  mpPacket->msgData.dataReq.msduLength = mi_envio;

1207  (void) MSG_Send(NWK_MCPS, mpPacket);

1212 }

```

Figura 6.3 Envío del parámetro ID del Meta-TEDS.

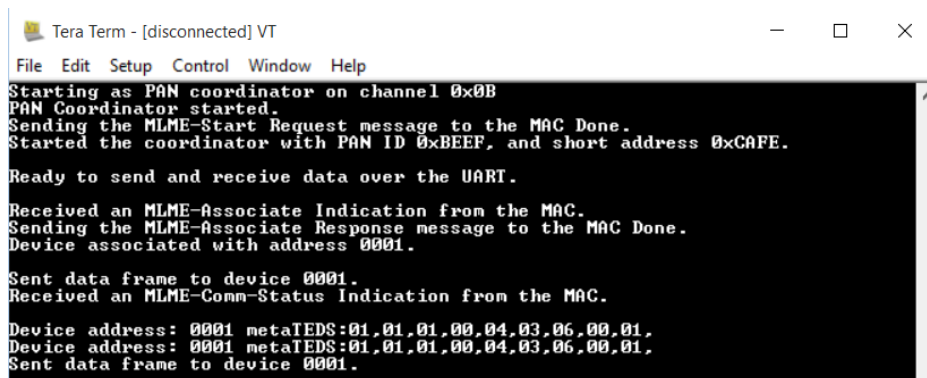


Figura 6.4 Meta_TEDS en NCAP.

de la variable *pMsdu* y los pone en la variable *mymeta.Id*, tal y como se muestra en la línea 973 de la Fig. 6.5.

```

902 static void App_HandleMcpsInput(mcpsToNwkMessage_t *pMsgIn)
903 {
904  int TEDSID_ok;

971  UartUtil_PrintHex((uint8_t*)&pMsgIn->msgData.dataInd.pMsdu[0], 9,
                    gPrtHexCommas_c);
973  TEDSID_ok =
        TEDSSetTedsId((uint8_t*)&pMsgIn->msgData.dataInd.pMsdu[0], &mymeta.Id);

990 }

```

Figura 6.5 Envío del parámetro ID del Meta-TEDS.

6.4. Resumen

Haber implementado el estándar IEEE 1451 con los módulos de *Freescale* y la biblioteca de Jan Popelka, permitió implementar el esquema de la Fig. 2.3. Para fines explicativos se presenta nuevamente en la Fig. 6.6.

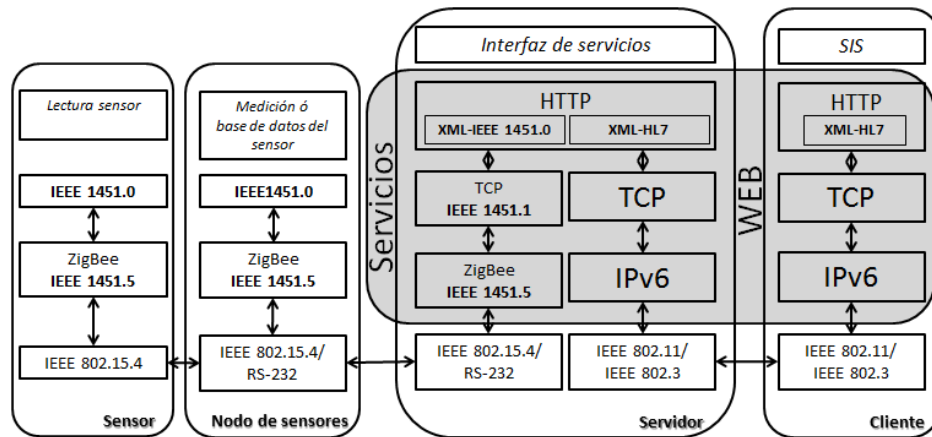


Figura 6.6 Interfaces IEEE 1451.x - HL7.

Donde la implementación de los estándares queda de la siguiente manera:

- Sensor ó *TIM*- IEEE 1451.0(TEDS) e IEEE 1451.5
- Nodo de sensores ó *NCAP* - IEEE 1451.0(TEDS/XML) e IEEE 1451.5
- Servidor (*Gateway*) - IEEE 1451.0 (XML), IEEE 1451.1 y HL7 (XML)

Se pudo implementar la codificación XML para mensajes en el *NCAP* con dispositivos dedicados, como lo es el MC1322x. Incluso es posible implementar el XML en el sensor, pero no se llevó a cabo por que el estándar IEEE 1451.x así los define. El *Gateway* es posible implementarlo en un módulo con circuiteria y programación dedicada, prescindiendo de la PC como en los trabajos relacionados.

Capítulo 7

Conclusiones

Dada la importancia de soluciones para el cuidado de la salud en pacientes de edad avanzada con enfermedades crónicas. Las tecnologías y estándares de comunicación actuales, ofrecen soluciones que pueden mejorar significativamente la prestación de atención médica de una manera que complementa y mejora los servicios existentes, mejorando así la calidad de vida del paciente. Tal es el caso de éste trabajo, el cual aspira a ser un elemento clave en beneficio de las personas en aquellos lugares que cuenten con tecnologías de sensores inalámbricos. Para ello se implementaron, en el esquema de salud de la Fig. 2.2, los estándares IEEE 1451.0, IEEE 1451.5 e IEEE 1451.5 del lado de los sensores. La familia de estándares IEEE 1451 es la solución del Instituto Nacional de Estándares y Tecnología (*NIST* - por sus siglas en inglés), a la necesidad de hacer interoperables sensores con diferentes estándares industriales. No obstante, y precisamente por su característica de interoperabilidad, lo hace atractivo para otras áreas de aplicación como lo es el área de la medicina, en éste caso, a través del estándar HL7. Con este trabajo, se corroboran los trabajos publicados en este campo. Y es parte medular en la respuesta a la pregunta de investigación: *¿Cómo disponer de la tecnología de la información y dispositivos inteligentes actuales, para la interoperabilidad de los servicios de e-Health?*. Los otros componentes son: los nodos de sensores fisiológicos colocados en los pacientes, nodos coordinadores para éstos sensores y que juntos formarán la red inalámbrica de sensores, un *Gateway* a los protocolos IEEE 1451-HL7 con capacidad de red local, un servidor local y un servidor médico. De la revisión de los trabajos relacionados con el *Gateway* IEEE 1451-HL7, se encontró que:

1. Para IEEE 1451.

- a) Los trabajos están dirigidos a sensores industriales.
- b) Han sido implementados en micro-controladores de 8 bits.

2. Para HL7, emplean el estándar IEEE 11073, el cual aborda la interoperabilidad de dispositivos médicos, pero no puede formar redes de sensores como lo hace el IEEE 1451.

7.1. Cumplimiento de los objetivos

El cumplimiento del objetivo general de ésta tesis, se llevó a cabo cumpliendo los siguientes objetivos específicos:

1. Se implementaron los TEDS de acuerdo a los estándares: IEEE 1451.0 y IEEE 1451.5, a través del IEEE 802.15.4, en el módulo *TIM*. Capítulo 6.
2. Se implementaron los TEDS de acuerdo a los estándares: IEEE 1451.0, IEEE 1451.1 y IEEE 1451.5, a través del IEEE 802.15.4, en el módulo *NCAP*. Capítulo 6.
3. La verificación de transmisión en éste estándar a través de la aplicación *Tera term*. Capítulo 6.
4. Los lineamientos e indicaciones específicas para el ó los ajustes al programa fuente para el cambio del protocolo IEEE 1451.0 al HL7. Capítulo 6.

7.2. Aportaciones

La principal aportación del presente trabajo fué demostrar y documentar la realización de la Fig. 1.3 a través de la Fig. 5.1

1. Implementación de la Biblioteca de Jan Popelka en micro-controladores modernos de 32 bits, lo que permitió implementarla también en el módulo *NCAP*. Jan Popelka usa una PC para estas tareas.
2. Implementación del protocolo IEEE 1451.5, a través del IEEE 802.15.4, para la comunicación entre los módulos *TIM* y *NCAP*. Jan Popelka implementó su biblioteca con protocolo IEEE 1451.4 para tal fin.

7.3. Trabajo futuro

Con el presente trabajo se comprueba la factibilidad del estándar IEEE 1451 para sensores inalámbricos. Las pruebas realizadas, sólo contemplaron la comunicación entre un módulo *TIM* y un módulo *NCAP* con dispositivos de desarrollo de *Freescale*. Por lo que se propone el siguiente trabajo a futuro:

1. Implementación de cualquiera de los estándares considerados por IEEE 1451.5 *id. est.* IEEE 802.11™, IEEE Bluetooth™, IEEE ZigBee™ y 6LoWPAN.
2. Implementación del estándar en más de dos módulos *NCAP* con más de dos módulos *TIM* en cada *NCAP*. Y probar las configuraciones como lo indicado en la Fig. 3.3 y la Fig. 3.4.
3. Implementación del estándar IEEE 1451 en el módulo *TWR-K60* de *Freescale*.
4. Lectura de otras variables fisiológicas.
5. Implementar actuadores.

7.4. Publicaciones

Se publicó y presentó el trabajo, el cual también aparece en la página de *IEEE Xplorer Digital Library*. Hasta el momento, cuenta con una cita [21], documento sometido al *Journal Medical System*, actualmente en revisión.

Ramírez-Ramírez Raúl, Cosío-León María, Ojeda-Carreño Dolores, Vázquez-Briseño Mabel, Nieto-Hipólito Juan Iván, “**Designing a Gateway IEEE1451-HL7 for E-health Telemonitoring Services**”, ICCSAT 2015, Octubre, Xalapa, Ver., DOI: 10.1109/ICCSAT.2015.7362927

Nomenclatura

Acrónimos

APL *Application Layer* - Capa de aplicación.

ARM *Architecture RISC Machine*

ASCCI *American Standard Code for Information Interchange* - Código Estándar Estadounidense para el Intercambio de Información.

CI *Circuit Integrated* - Circuito integrado.

CSMA-CA *Carrier Sense Multiple Access with Collision Detection* - Acceso múltiple con escucha de portadora y detección de colisiones.

Dot5AR *Dot 5 Approved Radio* - Punto 5 Aprobación de Radio.

ED *Energy Detector* - Detector de energía.

FFD *Full Function Device* - Dispositivo de funcionalidad completa.

GPIO *General Purpose Input/Output* - Entrada/Salida de Propósito General.

GTS *Guaranteed Time Slot* - Ranura de tiempo garantizada.

HTTP *Hypertext Transfer Protocol* - Protocolo de transferencia de hipertexto.

IDE *Integrated Development Environment*

IDE *Integrated Development Environment* - Ambiente de Desarrollo Integrado.

INEGI *Instituto Nacional de Estadística y Geografía*.

JTAG *Joint Test Action Group*.

KBPS *Kilo Bytes Per Second* - Miles de bytes (grupo de ocho bits) por segundo.

LCD *Liquid Crystal Display* - Pantalla de cristal líquido.

LQI *Link Quality Indicator* - Indicador de calidad de enlace.

LR-WPAN *Low Rate - Wireless Personal Area Network* - Redes inalámbricas de área personal con baja tasa de transmisión de datos.

LR-WPAN *Low Rate Wireless Personal Area Network* - Red inalámbrica de área personal con baja tasa de transmisión.

MAC *Medium Access Control* - Control de acceso al medio.

LAN *Local Area Network* - Red de área local.

MAN *Metropolitan Area Network* - Red de área metropolitana.

NCAP *Network Capable Application Processor* - Aplicación con proceso para capacidad de red.

NIST *National Institute of Standards and Technologies* - Instituto Nacional de Estándares y Tecnología

OSI *Open System Interconnection* - Modelo de interconexión de sistemas abierto.

PAN *Personal Area Network* - Red de área local.

PAN *Personal Area Network* - Red de área personal.

PC *Personal Computer* - Computadora Personal.

PDA *Personal Digital Assitant* - Asistente Digital Personal.

PHD *Personal Health Device* - Dispositivo de salud personal.

PHY *Physical* - Referente a la capa Física del modelo OSI.

PiP *Platform in Package* - Plataforma contenida en el paquete.

POS *Personal Operating Space* - Espacio de operación personal.

RAN *Regional Area Network* - Red de área regional.

RFD *Reduction Function Device* - Dispositivo de funcionalidad reducida.

RFID *Radio Frequency IDentification* - Idendificador por radio frecuencia.

RPM *Remote Patient Monitor* - Monitoreo de pacientes a distancia.

SI Sistema Internacional de medidas.

SMAC Simple MAC

SRFID *Sensor and Radio Frequency IDentification* - Sensor e idendificador por radio frecuencia.

TEDS *Transducer Electronic Data Sheet* - Datos del transductor en formato electrónico.

TIM *Transducer Interface Module* - Módulo de interfáz del transductor.

UART *Universal Asynchronous Receiver-Transmitter* - Transmisor-Receptor Asíncrono Universal.

UUID *Universal Unique Identification Data Type Structure* - Estructura de datos para la identificación universal única.

WPAN *Wireless Personal Area Network* - Red inalámbrica de área personal.

XML *eXtensible Markup Language* - Lenguaje de marcas extensible.

Bibliografía

- [1] IEEE 1451.0. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats*. The Institute of Electrical and Electronics Engineers, Inc, New York, 2007.
- [2] IEEE 1451.1. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Network Capable Application Processor (NCAP) Information Model*. The Institute of Electrical and Electronics Engineers, Inc, New York, 1999.
- [3] IEEE 1451.2. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*. The Institute of Electrical and Electronics Engineers, Inc, New York, 1998.
- [4] IEEE 1451.3. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems*. The Institute of Electrical and Electronics Engineers, Inc, New York, 2004.
- [5] IEEE 1451.4. *IEEE Standard for A Smart Transducer Interface for Sensors and Actuators-Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*. The Institute of Electrical and Electronics Engineers, Inc, New York, 2004.
- [6] IEEE 1451.5. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators-Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*. The Institute of Electrical and Electronics Engineers, Inc, New York, 2007.
- [7] IEEE 1451.7. *IEEE Standard for Smart Transducer Interface for Sensors and Actuators-Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats* . The Institute of Electrical and Electronics Engineers, Inc, New York, 2010.
- [8] IEEE 802.15.4. *IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. The Institute of Electrical and Electronics Engineers, Inc, New York, 2011.
- [9] ABIresearch. Wearable sports and fitness devices will hit 90 million shipments in 2017, February 2012. URL <https://www.abiresearch.com/press/wearable-sports-and-fitness-devices-will-hit-90-mi/>. [online] <https://www.abiresearch.com/press/wearable-sports-and-fitness-devices-will-hit-90-mi/>.

- [10] Manabrenda Bhuyan. *Intelligent Instrumentation: Principles and Applications*. CRC Press, Boca Raton, FL., 2011.
- [11] Dolores Ojeda Carreño. Método para la integración de datos provenientes de sensores fisiológicos basado en los estándares hl7 para un sistema médico de monitorización a distancia. Master's thesis, Universidad Autónoma de Baja California, Facultad de Arquitectura, Ingeniería y Diseño, 2012.
- [12] Konstantin Mikhaylov et. al. Ieee 1451: A standard in support of smart transducer networking. In *Intelligent Sensor Networks: Across Sensing, Signal Processing, and Machine Learning*, pages 55–76, 2012. doi: 10.1201/b14300-5.
- [13] Fei Hu and Qi Hao, editors. *Intelligent Sensor Networks The Integration of Sensor Networks, Signal Processing and Machine Learning*. Sensor, Instrumentation & Measurement. CRC Press, Boca Raton, FL, 2012.
- [14] IAR 2009-2012. Getting started with iar embedded workbench®, April 2012.
- [15] IEEE 1451.6. Draft standard for a smart transducer interface for sensors and actuators-a high speed canopen based transducer network interface for intrinsically safe and nonintrinsically safe applications, 2016. URL <http://grouper.ieee.org/groups/1451/6/>.
- [16] INEGI. Número de habitantes, 2015. URL <http://cuentame.inegi.org.mx/poblacio/habitantes.aspx?tema=P>.
- [17] Jaehwan Kim, Jeongil Heo, and Wooshick Kim. A study on the implementation of ieee1451 for e-health. *IEEE Transactions*, 1998.
- [18] Wooshik Kim, Suyoung Lim, Jinsoo Ahn, Jiyoung Nah, and Namhyun Kim. Integration of ieee 1451 and hl7 exchanging information for patient's sensor data. *Journal of Medical Systems*, page 9, 2010. doi: 10.1007/s10916-009-9322-5. URL <http://dx.doi.org/10.1007/s10916-009-9322-5>.
- [19] Kang Lee. A smart transducer interface standard for sensors and actuators. In Zurawski R., editor, *The Industrial Information Technology Handbook*, pages 1103–1118. CRC Press, 2004. URL http://www.iestcfa.org/books/iit_lee.htm.
- [20] Malrey Lee and Thomas M. Gatton. Wireless health data exchange for home healthcare monitoring systems. *sensors*, page 18, 2010. ISSN 1424-8220. doi: 10.3390/s100403243. URL <http://www.mdpi.com/1424-8220/10/4/3243>.
- [21] Cosío-León María, Ojeda-Carreño Dolores, Ibarra-Hernández José Alberto, and Nieto-Hipólito Juan Iván. A practical approach to semantic interoperability using hl7 v3 and ieee1451 standards into embedded devices. In *Journal Medical System*, pages 1–6, 2016. Under revision.
- [22] Bridget Moorman. Medical device interoperability: Standards overview. *Biomedical Instrumentation & Technology*, pages 132–138, 2010.
- [23] Jan Popelka. *Komunikace senzorů pomocí standardu IEEE1451*. PhD thesis, České vysoké učení technické v Praze, Fakulta elektrotechnická Katedra měření, 2010.

- [24] Jan Popelka. Performance of smart sensors standards for aerospace applications. *Electrotechnical Review*, page 4, 2012. ISSN 0033-2097. URL <http://pe.org.pl/articles/2012/1a/49.pdf>.
- [25] Richard L. Fischer, Jeff Burch. The picmicro® mcu as an ieee 1451.2 compatible smart transducer interface module (stim). Tech Note, September 2000. URL <http://ww1.microchip.com/downloads/en/AppNotes/00214a.pdf>. .
- [26] uC003. The aduc812 microconverter as an ieee 1451.2 compatible smart transducer interface. Tech Note, September 1999. URL http://www.analog.com/media/en/technical-documentation/application-notes/uC003___The_ADuC812_as_an_IEEE_1451.2_STIM.pdf. .
- [27] Y. Eugene Song. An open implementation of ieee 1451, April 2013. URL https://sourceforge.net/projects/open1451/?source=typ_redirect. .