

Diseño de un nodo con capacidad plug & play

Gallina, Sergio Hilario (*Author*)

Departamento Electrónica
Facultad de Tecnología y Ciencias Aplicadas-UNCa
Catamarca, Argentina
sgallina@tecno.unca.edu.ar

Beltramini, Paola (*Author*)

Departamento Electrónica
Facultad de Tecnología y Ciencias Aplicadas-UNCa
Catamarca, Argentina
pbeltramini@tecno.unca.edu.ar

Villagran, Daniel (*Author*)

Departamento Electrónica
Facultad de Tecnología y Ciencias Aplicadas-UNCa
Catamarca, Argentina
dvillagran@tecno.unca.edu.ar

Ferraro Matias (*Author*)

Estudiante de Ingeniería Electrónica
Facultad de Tecnología y Ciencias Aplicadas-UNCa
Catamarca, Argentina
matiasferraro@yahoo.com

Arjona, Lucas (*Author*)

Estudiante de Ingeniería Electrónica
Facultad de Tecnología y Ciencias Aplicadas-UNCa
Catamarca, Argentina
lucas_arjona_88@hotmail.com

Lobos, Diego (*Author*)

Estudiante de Ingeniería Electrónica
Catamarca, Argentina
Facultad de Tecnología y Ciencias Aplicadas-UNCa

Abstract— El presente trabajo plantea el desarrollo de un nodo estandarizado, con capacidad no solo para interactuar con otros nodos sino también para controlar uno o mas transductores en tiempo real. El objetivo es poder transferir datos de control a todos los componentes de una red ubicua, considerando que es necesario poseer un sistema que elimine los problemas de los dispositivos aislados. De este modo resulta seguro que los componentes del sistema se comuniquen a través de un lenguaje común. El desarrollo se basa en el estándar IEEE 1451; los resultados logrados son alentadores y se fundamentan en el comportamiento del prototipo y la simulación, donde se han probado las funcionalidades básicas de un nodo y su interacción con otros nodos de la red. La aplicación de estos nodos ha sido orientado a instalaciones domóticas y para sistemas de riego agroindustriales.

Keywords— **Nodo Plug & Play; NCAP; STIM; Transductor Inteligente**

I. INTRODUCCIÓN

Las normas IEEE 1451 definen un conjunto de interfaces de comunicación, comunes para estandarizar la conectividad de los transductores a los microprocesadores y las redes. Esto se logra mediante la definición de diversos aspectos que participen del desarrollo de transductores inteligentes en red.

Debido a la diversidad de dispositivos en el mercado y de las aplicaciones donde estos se incluyen, se propone aquí el desarrollo de un nodo estandarizado con características Plug & Play que puede ser aplicado a cualquier transductor, para lograr su integración a una red y que además responda a los estándares de la familia IEEE 1451 [1], [2], [3] Esta familia de normas está diseñada para facilitar la conectividad de los sensores y actuadores a un dispositivo o red. Éstos, por su parte, se conectan con un módulo de interfaz de transductor inteligente (STIM), que se indica *inteligente* ya que

proporciona funciones, como la auto-identificación, auto-calibración, etc. El STIM está controlado por un procesador de aplicaciones con capacidad de conectarse a una red (NCAP) en un entorno independiente. Dependiendo de los STIMs, uno o varios se pueden conectar a un solo NCAP. Cada STIM define una Hoja de Datos Electrónica del Transductor (TEDS) y una interfaz de comunicación para la conexión con un NCAP. Las TEDS permiten la auto-identificación de los transductores, lo que posibilita la conexión plug & play; a la vez la interfaz de comunicación permite que el NCAP pueda acceder y controlar el STIM.

II. DEFINIENDO EL PROBLEMA

Desde hace varios años se desarrollan soluciones para integración de sistemas y equipos. En la arquitectura de una aplicación de este tipo, se distinguen tres capas:

1. Capa de dispositivos (hardware): En esta capa se encuentran los sensores que permiten recoger información, y los actuadores que permiten interactuar con el entorno.
2. Capa correspondiente a la plataforma (software): Esta capa a su vez se divide en:
 - *Interfaces de acceso a los dispositivos*: integra la información proveniente de los sensores, proveyendo una interfaz única a los servicios.
 - *Módulos de servicios*: integra módulos con funcionalidades desarrolladas por cada una de las empresas en sus áreas de conocimiento.
 - *Capa de coordinación y composición de servicios*: es la que permite que se desarrollen servicios autónomos, coordinando los diferentes módulos anteriores.

○ *Programación:* la plataforma es accesible mediante una API de alto nivel, que facilita la interacción y programación de la misma.

3. Capa de entorno de aplicación: aquí se encuentran todos los escenarios posibles de ser aplicación.

Teniendo en cuenta lo mencionado, se considera necesario avanzar en el desarrollo de sistemas convergentes que permitan adoptar soluciones fáciles y útiles, tendientes a la interacción entre diferentes dispositivos. A lo largo de estos últimos años han existido propuestas para la integración de dispositivos heterogéneos en redes ubicuas, propuestas van desde simples protocolos de comunicación e integración de dispositivos a plataformas más elaboradas, que no solamente posibilitan la comunicación entre las diferentes aplicaciones del entorno, sino que, además, añaden facilidades para su desarrollo y gestión.

En base a todo lo expuesto, consideramos necesario el desarrollo de nodos inteligentes que incorporen las tres capas mencionadas en forma transparente para los usuarios, que puedan actuar en forma cooperativa y con costos accesibles.

III. REQUERIMIENTOS

El diseño de un nodo de red que cumpla con el estándar IEEE 1451, debe incluir una serie de requisitos. Estos requisitos se pueden dividir en cuatro subsecciones:

1. Requisitos del sistema para el comportamiento específico de la aplicación.
2. Los requisitos NCAP para la funcionalidad de alto nivel.
3. Requisitos de STIM para la funcionalidad de nivel superior.
4. Estructura de un transductor inteligente capaz de manejar un dispositivo de tiempo real.

El requisito de diseño común entre los objetos del sistema es que se plantee en un entorno de red y transductor independiente. El diseño está centrado en el estándar IEEE1451.1 para la funcionalidad de la NCAP, mientras que para el funcionamiento de la STIM en el estándar IEEE 1451.2.

3.1 REQUISITOS DEL SISTEMA

Los requisitos del sistema son funcionales y no funcionales. Las normas no especifican los requisitos no funcionales ya que estos son definidos en la aplicación y puesta en práctica. Los requisitos funcionales abarcan el comportamiento del sistema según lo declarado por la familia IEEE 1451. El requisito principal es que el diseño debe ser independiente de la red y del transductor, lo que significa que a nivel de aplicación, las conexiones físicas en el sistema se vuelven transparentes para los usuarios.

Con respecto a la independencia de la red y del transductor, esta se logra mediante la implementación por bloques, diseñados e implementados como se muestra en la Fig. 1.

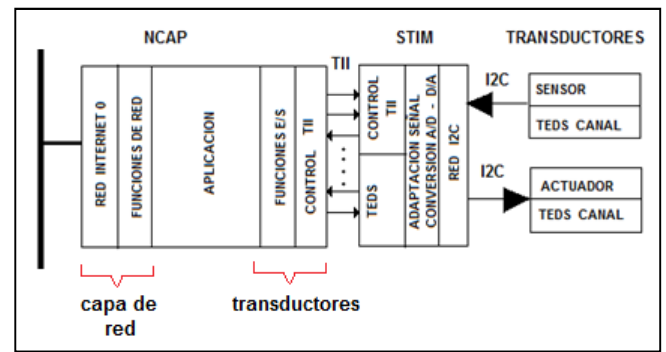


Fig. 1: Estructura de Diseño de Referencia

El hardware y las funciones de red están definidos como específicos en la norma para la implementación, es por eso que la funcionalidad no está especificada en el estándar, y depende de la aplicación. Sin embargo, estos bloques, deberán interactuar y cumplir con las especificaciones eléctricas de la red subyacente. Los bloques restantes se describen en la norma y se diseñarán siguiendo sus especificaciones.

3.2 NETWORK CAPABLE APPLICATION PROCESSOR (NCAP)

El NCAP consta de hardware y bloques de software compatibles con el estándar IEEE 1451.1. El NCAP se divide en tres capas:

- Capa de red, el hardware consiste en la lógica específica de la red necesaria para la red subyacente.
- Capa de aplicación consistirá únicamente de bloques de software, los que están compuestos por bloques estándar definidos para la aplicación, conjuntamente con el código específico para la aplicación.
- Capa de conexión con los transductores, el hardware consistirá en la Interfaz Independiente del Transductor (TII) definida en la IEEE 1451.2.

Es importante señalar, que el software para algunos bloques está definido por el estándar, mientras que otros son específicos de la implementación. Sin embargo, estos objetos se comunican entre sí utilizando la orientación de la norma 1451.1.

Los tres grandes bloques trabajarán conjuntamente para comunicarse con el "mundo real".

3.2.1 Capa de Red

Tiene la función de manejar las comunicaciones en la red. El hardware consiste en los bloques lógicos necesarios para la implementación del protocolo Internet Cero (I0) sobre la misma línea de alimentación eléctrica de los nodos. Las funciones de red, será la "biblioteca" que tendrá a su cargo la codificación y decodificación de datos hacia y desde el formato en el cable de la red.

Las comunicaciones se efectuarán utilizando los dos modelos de comunicación que se definen en la norma: cliente-servidor y editor-suscriptor. El primero de ellos se utiliza exclusivamente para la autoidentificación del nodo en la red; mientras que el segundo para la operación normal del nodo.

3.2.2 Capa de aplicación

Dentro de esta capa se define el comportamiento específico de la aplicación de la NCAP.

El comportamiento de la aplicación no está afectado por las diferentes conexiones físicas que se pueden realizar entre el NCAP/red y NCAP/TIM. La capa de aplicación también actuará como un puente entre las operaciones de la red y las operaciones de transductores. Se deberá tener un código específico, pertinente para el control particular o sistema de seguimiento que se implementa.

3.2.3 Capa de Transductor

La capa transductor contiene tanto hardware como módulos de software que manejan las comunicaciones del STIM con el NCAP. El bloque de hardware es responsable de la conexión física NCAP/STIM; este bloque debe cumplir con las especificaciones eléctricas del estándar IEEE1451.2.

El software es responsable de las interacciones con cualquier transductor que está conectado físicamente a la STIM. Esto se hace a través de las funciones de entrada o salida. Por lo tanto, este bloque será responsable de decodificación y codificación de la información que se envía y recibe por la comunicación NCAP/STIM.

Todos los comandos que son compatibles con el STIM están representados en esta capa. Una orden común que se representará en la capa transductor es el comando de disparo. Este comando permite al NCAP leer o ajustar los transductores que están conectados físicamente al sistema.

3.3 SMART TRANSDUCER INTERFACE MODULE (STIM)

El STIM contiene una combinación de hardware y software que mantienen la funcionalidad de un transductor inteligente. Para que el STIM se considere "inteligente", deberá tener la capacidad de auto-identificación ante el NCAP, generar interrupciones y poseer las hojas electrónicas de datos o TEDS. Otra funcionalidad que tiene el STIM será la capacidad de comunicarse con un NCAP para manejar la activación, generar interrupciones además de establecer la interfaz con los transductores físicos.

El STIM se inicializa (por el NCAP o sólo) después que se ha encendido, y entra en estado operativo hasta que se reinicie el NCAP o se apaga. No requiere fuente de alimentación, ya que ésta es suministrada por el NCAP.

3.3.1 Hoja Electrónica de Datos (TEDS)

Las TEDS son el núcleo de la funcionalidad del STIM, ya que proporciona la funcionalidad de plug & play mediante la implementación de la función de auto-identificación de los transductores. Este bloque puede ser generado en tiempo de fabricación o de forma remota a través de un NCAP.

Los TEDS residen en la memoria no volátil y describen completamente el STIM mediante la Meta-TEDS que describe el STIM en su totalidad. La información que se representa en la Meta-TEDS incluye la cantidad de transductores que se implementan, la velocidad de muestreo máxima y mínima del sistema y una suma de comprobación de integridad de datos.

Existe también un bloque etiquetado de TEDS del canal, que describe cada transductor conectado al STIM. La TEDS de información del canal incluye el tipo de transductor, la unidad física del transductor, el modelo de datos, la información de tiempo (tiempo de actualización, escritura, de preparación, periodo de muestreo) y una suma de comprobación de integridad de datos.

Otros bloques TEDS opcionales, no implementados en este trabajo, tienen información tal como la calibración del transductor.

3.3.2 Registro de Estado

El STIM y los transductores inteligentes, disponen de registros de estado tanto estándar como auxiliar. Los registros implementados en esta aplicación son los registros estándar del STIM y los registros estándar de cada canal. A ellos se accede mediante la ejecución de comandos: *leer el estado del canal estándar* para el canal en cuestión o el comando *leer el estado global estándar* para el STIM.

El estado devuelto será de 2 bytes y el significado de cada bit se muestra en la Tabla 1. Los bits marcados como OR implementan una función lógica OR de los bits respectivos de cada canal

IV. DISEÑO DEL NODO

El diseño del nodo se realiza en forma totalmente modular, mediante una combinación de hardware y software.

Comenzamos el diseño mediante la adopción del protocolo *Internet 0* sobre la línea de alimentación de los nodos, como la capa de red a utilizar para la interconexión de los mismos. Se adopta el modelo de interface propuesto por la IEEE 1451.2 para la comunicación NCAP/STIM, el mismo consiste en una conexión serial basada en 10 hilos o señales. Además se emplea el protocolo I²C para conexión de los diferentes transductores que pueden incorporarse al nodo. El modelo adoptado se muestra en la Fig. 2.

TABLA 1: BIT DE LOS REGISTROS DE ESTADO ESTÁNDAR PARA EL STIM (GLOBAL) Y PARA LOS CANALES TRANSDUCTORES

bit	STIM (Global)	Canal Transductor
msb	Libre	Libre
14	Libre	Libre
13	Libre	Libre
12	Libre	Libre
11	Reservado	Reservado
10	Reservado	Reservado
9	Reservado	Reservado
8	STIM operativo	Canal operativo
7	STIM bit de error (OR)	Error de hardware del canal
6	Dato/Evento en el canal (OR)	Dato/Evento en el canal
5	Dato / Evento perdido (OR)	Dato / Evento perdido
4	Registro de estado auxiliar (OR)	Registro de estado auxiliar
3	STIM comando inválido	Reservado
2	STIM existió un reset	Reset del canal
1	STIM reconocimiento de disparo	Reconocimiento de disparo
lsb	Solicitud de servicio	Solicitud de servicio

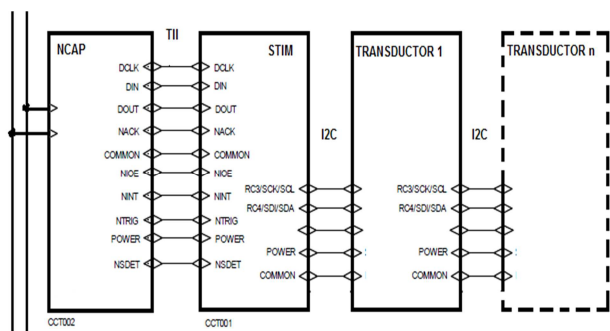


Fig. 2: Diagrama de bloques del nodo con transductores

De esta forma un nodo se conecta a una única STIM y a través de esta, a un máximo de 255 transductores. Este límite teórico es muy superior a las necesidades de una instalación domótica.

4.1 NCAP

Como ya se explicó anteriormente, este módulo consta de tres capas, la primera de ellas, la capa de red, implementa el Protocolo Internet Cero sobre la línea de alimentación del nodo [7].

En la capa de aplicación, nos encontramos con el software de control del NCAP y de la propia aplicación del nodo, la primera responde al estándar IEEE 1451.1 y la segunda a la función del nodo dentro de una aplicación domótica.

Los bloques de software están diseñados en C, que no es un lenguaje de programación orientado a objetos; esto presenta algunos problemas en el diseño de la jerarquía de clases y relaciones de objeto propietario. Para resolver este problema, los objetos están diseñados como estructuras y la jerarquía de clase está diseñada mediante la definición de una súper clase que actúa como *padre*.

En la Tabla 2 se describen los objetos de software del NCAP que son responsables de las comunicaciones de red utilizando el modelo cliente/servidor (para la configuración-CP) o el modelo editor/suscriptor (Servicios en tiempo real-TR).

TABLA 2: PRINCIPALES FUNCIONES DEL SISTEMA

Función	Interfaz	Definición
<i>Solicita_Inclusion()</i>	CP	Nodo esclavo solicita su inclusión en la Red
<i>configura ()</i>	CP	Nodo maestro envía parámetros de configuración a un nuevo nodo
<i>Reconfigura ()</i>	CP	Nodo maestro envía una nueva configuración de parámetros .
<i>Detener ()</i>	CP	Nodo maestro saca de operación a un esclavo nodo.
<i>Iniciar ()</i>	CP	Nodo maestro pone operativo a un esclavo nodo.
<i>Test_Transductor ()</i>	CP	Nodo maestro verifica si un nodos está activo.
<i>Transmite ()</i>	TR	Nodo envía un paquete de datos a la red.
<i>Diagnóstico ()</i>	TR	Diagnóstico de un sensor inteligente.

4.1.1 Comunicación modelo Cliente - Servidor

El modelo *cliente-servidor* permite la comunicación entre los grupos de nodos clientes y un nodo servidor. Este

enfoque es muy eficiente cuando los datos se encuentran en un servidor central. Sin embargo, este modelo es considerado ineficaz cuando los datos son producidos por muchos nodos y son consumidos por otros muchos nodos, propios de entornos distribuidos [4].

En nuestro caso se utiliza este mecanismo sólo para la autenticación de un nuevo nodo en la red. En la Fig. 3 se muestra el diagrama de secuencias de este proceso

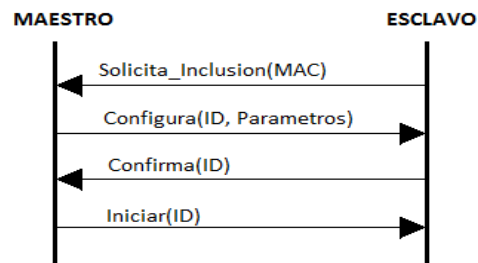


Fig. 3: Diagrama de secuencias de un proceso de autenticación de un nuevo nodo en la red

4.1.2 Comunicación modelo Editor - Suscriptor

El modelo de intercambio de datos *editor-suscriptor*, permite el intercambio de la mensajería síncrona y asíncrona (eventos) entre dispositivos. Este modelo ofrece un alto grado de desacoplamiento entre los nodos, ya que cada nodo de la red no tiene la necesidad de conocer la existencia de otros nodos.

El modelo *editor-suscriptor* define dos tipos de entidades: *los editores* y *los suscriptores*. Los primeros (*editores*) son responsables del envío de mensajes (comienza el comunicación). Los suscriptores (*usuarios registrados*) son los nodos que consumen estos mensajes.

El mensaje se transmite (*broadcast*) a través de la red; al recibir un mensaje nuevo, los suscriptores revisan el tipo y analizan si es parte de lista de temas de interés. Si el nodo está interesado en el mensaje, se procesa, de lo contrario, simplemente se descarta.

En el modelo editor-suscriptor en tiempo real se definen los siguientes parámetros para los mensajes de publicación [5]:

Tema: identifica de forma única una publicación específica;

Tipo: identifica el formato de datos de una publicación ;

Prioridad: permite establecer el peso relativo de los mensajes del mismo tema con el fin de posibilitar el arbitraje en el envío;

Persistencia: tiempo de validez de una publicación.

Separación mínima entre las publicaciones consecutivas: ninguna publicación nueva de un mismo tipo se acepta hasta que un tiempo mínimo haya transcurrido;

Plazo (deadline): el plazo máximo entre la recepción de dos publicaciones consecutivas del mismo tipo.

4.2 STIM

Para satisfacer las funciones indicadas, se ha implementado una estructura simple que se muestra en la Fig. 4. La implementación de las TEDS, el registro de estado y la máscara de interrupciones, se han realizado sobre una

memoria EEPROM externa; los demás componentes, se implementaron en un microcontrolador PIC.

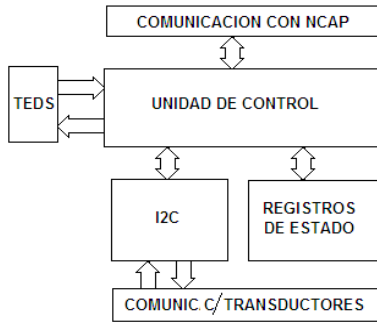


Fig. 4: Estructura general STIM

Uno de los aspectos clave de esta arquitectura es que va a permitir un número variable de transductores. Por lo tanto, la configuración particular a los diferentes artefactos hogareños se puede realizar con un mínimo esfuerzo de programación.

4.2.1 Hoja electrónica de datos (TEDS)

En el STIM se implementa la META-TEDS, su estructura está fundamentada en el estándar IEEE 1451.2.

Un acceso a esta estructura pondrá a disposición de otros nodos toda la información necesaria para el acceso a cualquier canal, además de la información común a todos los canales.

Por razones de extensión del trabajo no se detalla la estructura de esta TEDS pero es importante mencionar que en ella se debe especificar el campo UUID o Identificador único universal. El UUID debe garantizarse como único, será de 10 bytes de longitud y constará de cuatro subcampos, a saber: ubicación, fabricante, año, y tiempo.

Además de esta TED el STIM se ha diseñado para almacenar una copia de las TEDS de los canales a él conectados, esta copia se actualiza en el momento del reconocimiento de un canal que se conecta al STIM. De esta forma se reducen las transacciones entre STIM y Canal

4.2.2 Comunicación NCAP/STIM

Una dirección completa será de 2 bytes de largo y estructurado de la siguiente forma: desde el bit 0 al bit 7 tendremos el número del canal al cual dirigimos el mensaje, si esta dirección es cero el mensaje se dirige al STIM y se considera comando del tipo global ya que afecta a todos los canales. Los bits 8 a 15 definen la dirección funcional o comando, aquí el bit 15 se utiliza para identificar si el comando es de lectura ("1") o de escritura ("0").

Las direcciones funcionales y del canal, son direcciones lógicas. El mapeo de direcciones funcionales o canal, a las direcciones físicas se realiza dentro de la STIM y forma parte del software de base de la aplicación. La dirección funcional está subordinada a la dirección del canal.

A cada transductor en un STIM se le asignará un número de canal. Un STIM puede tener hasta 255 canales.

La Tabla 3 resume la lista de comandos básicos implementados.

TABLA 3: COMANDOS DE TIM

Dirección Funcional	Comando
0	Escribir datos en los transductores
1	Escribir comando de control
3	Escribir un disparo a un canal
5	Escribir la máscara de interrupciones
128	Leer datos de transductores
130	Leer el registro de estado
160	Leer TEDS
161	Leer TEDS de identificación (solo para canales)

4.2.3 Registro de estado e Interrupciones

El registro de estado del STIM se implementó con cinco bits, estos son: *STIM Operativo*, *Reconocimiento de disparo*, *Comando Inválido*, *Peticion servicio Global*, *Canal auxiliar disponible*.

No se han implementado los registros de estado auxiliares y tampoco se ha hecho uso de los bits de libre disponibilidad, los cuales se esperan utilizar al completarse el diseño de una aplicación específica del nodo.

Las interrupciones se generan por una combinación del registro de estado y la máscara de interrupciones. Con este esquema el bit de interrupciones del protocolo TII es un OR lógico de todas las operaciones AND en la combinación con la máscara de interrupciones /estado, así que cuando este bit se pone en "1" se genera una interrupción al NCAP. De esta forma una interrupción puede tener 16 diferentes orígenes. El NCAP deberá poseer un controlador de interrupciones con el fin de procesar las peticiones.

4.3 TRANSDUCTOR INTELIGENTE

En la Fig. 5 se muestra el diagrama en bloques de un sensor inteligente desarrollado en el marco del proyecto para detectar temperatura y humedad ambiente [8], a excepción del acondicionador de señal y del oscilador el resto de los bloques se ha desarrollado sobre un microcontrolador PIC.

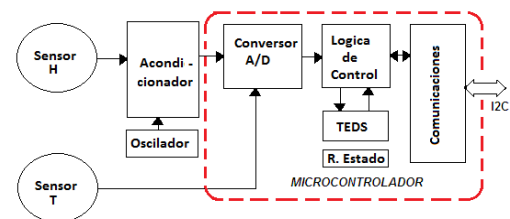


Fig. 5: Diagrama en bloques del transductor inteligente

4.3.1 Registro de estado

El estado actual del transductor (Tabla 1) está representado de tal manera que es accesible mediante los comandos correspondientes. Para los canales individuales, los bits de estado implementados en el prototipo, son: *Reconocimiento de disparo*, *Canal Operativo*, *Peticion de servicio del canal* y *Canal auxiliar disponible*.

4.3.2 Las TEDS del canal transductor

Las TEDS del sensor fueron implementadas en la memoria EEPROM del MCU y contiene todas las particularidades del canal y su estructura responde al estándar IEEE 1451.0

Para el sensor se implementaron dos TEDS, la *Transducer Channel TEDS* y la *User's Channel Name TEDS* (TEDS Nombre del Canal)

4.3.2.1 Transducer Channel TEDS

Es una TEDS obligatoria para el estándar y aporta información detallada sobre el transductor (Tabla 4).

TABLA 4: TEDS DEL CANAL "SENSOR DE HUMEDAD"

Campo:	Descripción	Valor
LENGTH	Largo, incluye el checksum	00-00-00-5E
TEDSID	Identificación	03-04-00-03-01-01
CALKEY	Capacidad de calibración	0A-01-01
CHANTYPE	Tipo de canal (0, sensor)	OB-01-00
PHYUNITS	Nombre de la unidad física a medir.	0C-09-32-01-01-35-01-7A-36-01-7A
LOWLIMIT	Límite inferior (10%)	0D-04-41-20-00-00
HILIMIT	Límite superior (100%)	0E-04-42-C8-00-00
OERROR	Incertidumbre típica (2%)	0F-04-40-00-00-00
SELFTTEST	Auto test (sin implementar)	10-01-00
SAMPLE	Modelo de datos de la señal muestreada: 10 bits	12-09-28-01-00-29-01-02-2A-01-0A
UPDATET	Tasa de actualización (10 muestras/seg. - 100 ms)	14-04-3D-CC-CC-CD
RSETUPT	Tiempo de preparación (25 µs)	16-04-37-D1-B7-17
SPEIOD	Tiempo que demora el sensor (100ms)	17-04-3D-CC-CC-CD
WARMUPT	Demora en estabilizarse (5 S)	18-04-40-A0-00-00
RDELAYT	Tiempo entre 2 comandos de lectura (100 mSeg)	14-04-3D-CC-CC-CD
SAMPLING	Atributos del muestreo (libre)	1F-03-30-01-02
CHECKSUM	Suma de comprobación	F0-72

Esta TEDS se mapea en la memoria EEPROM interna del microcontrolador ocupando las direcciones de 0 a 98. Ocupa 99 Bytes de memoria en total.

4.3.2.2 User's Transducer Name TEDS

Es una TEDS obligatoria para el estándar y almacena el nombre por el cual se reconocerá al sensor en una aplicación hacia el usuario.

TABLA 5: TEDS "NOMBRE DEL CANAL SENSOR DE HUMEDAD"

Campo	Descripción	Valor
LENGTH	Largo, incluye el checksum	00-00-00-16
TEDSID	Identificación	03-04-00-0C-01-01
FORMAT	Define el formato de campo datos	04-01-00
TCName	Nombre del sensor "SHH-1 v1.0"	53-48-48-2D-31-20-76-31-2E-30
CHECKSUM	Suma de comprobación	FC-C5

Esta TEDS se graba en la memoria EEPROM interna del microcontrolador a continuación de la TEDS anterior y ocupa las direcciones de 99 a 125 (Tabla 5). Ocupa 26 Bytes de memoria en total.

Como se puede observar, la información de las TEDS incluye lo descrito en el apartado 3.3.1.

4.3.3 Funcionamiento

El transductor podrá estar en estado 'operativo' ó 'no operativo', esta situación se establece según el estado del bit *canal operativo* en el registro de estado. En estado no operativo el transductor no realiza ninguna acción y espera el comando de desbloqueo.

El transductor en situación operativa podrá encontrarse en uno de tres estados estables, a) En el estado de

inicializando, se limpia el buffer de datos y se actualiza el registro de estado. La inicialización puede provenir por el encendido (*Power-On*) o por la llegada de un comando *Reset* desde el exterior; b) En el estado *libre* el transductor ya se ha identificado y se encuentra tomando muestras en forma libre, se ha previsto un buffer que almacene hasta un máximo de 10 muestras. c) Al estado *activo* se ingresa por la llegada de un comando seguido de una señal de disparo, si el comando recibido es válido se ejecuta y se establece la señal de reconocimiento de disparo, si el disparo es invalido se establece el bit de *datos/evento perdido*.

V. RESULTADOS

El desarrollo de transductores inteligentes se encuentra en fases iniciales pero con experiencias alentadoras. Para pruebas de laboratorio se implementaron prototipos obteniendo buenos desempeños en la transmisión y recepción de datos y comandos, interacción en red, etc.

VI. CONCLUSIONES

Se presentó el desarrollo de una arquitectura de nodo autónomo para el control de transductores inteligentes. La arquitectura tiene características basadas en las normas IEEE 1451.

Se utilizó el concepto de independencia de la infraestructura de la citada norma y se definieron las interfaces de comunicación entre nodos basado en el protocolo Internet 0 y de sensores inteligentes con el STIM mediante I2C.

Con los resultados obtenidos a través de los ensayos, concluimos el proceso de estudio y comprensión de redes ubicuas, donde cada nodo puede realizar su función en la red e interactuar con otros nodos para contribuir a un sistema mayor como podría ser una instalación domótica o un sistema de control de una producción agro-industrial.

VII. REFERENCIAS

- [1]. IEEE 1451.0 Draft Standard for a Smart Transducer Interface for Sensors and Actuators – Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats (January 2007)
- [2]. IEEE 1451.1 Standard for a Smart Transducer Interface for Sensors and Actuators-Network Capable Application Processor (NCAP) Information Model, June 26, 1999
- [3]. IEEE 1451.2 Standard for a Smart Transducer Interface for Sensors and Actuators-Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats, (September 26, 1997)
- [4]. Ocera, Wp2 - architecture specification. deliverable d2.1 - architecture and components integration.
URL: <http://mmis.fr/en/support/doc/architecture/> (2002)
- [5]. Dolejs, O., Petr S. & Zdenek H., On the ethernet use for realtime publish-subscribe based applications, 'Proceedings. IEEE International Workshop on Factory Communication Systems', pp. 39-44. (2004)
- [6]. Lopez, Gustavo Eduardo, System-on-a-chip solution for plug and play networked smart transducers. University of pittsburgh - school of engineering, (2004)
- [7]. Lobos, D. M. & Fugaza M. . "Implementación del Protocolo Internet Cero". Trabajo Final de Carrera. Ingeniería Electrónica. FT y CA, UNCa. (2013)
- [8]. Arjona, L. M. "Desarrollo de un prototipo de sensor inteligente de humedad en hoja para aplicación en riego de precisión". Trabajo Final de Carrera. Ingeniería Electrónica. FTyCA, UNCa. (2013)