



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de La Laguna

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

"Diseño y validación de un módulo e interfaz para
transductores inteligentes a través del protocolo

21451 de la IEEE"

POR

Ing. Carlos Antonio Olivas Pérez

T E S I S

PRESENTADO COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE MAESTRO
EN CIENCIAS EN INGENIERÍA ELÉCTRICA

DIRECTOR DE TESIS

Dr. Emmanuel Gómez Ramírez

ISSN: 0188-9060



RIITEC: (21)-TMCIE - 2019

Torreón, Coahuila. México

Diciembre 2019



"2019, Año del Caudillo del Sur, Emiliano Zapata"

Torreón, Coah., 04/Noviembre/2019
Dependencia: DEPI/CPCIE
Oficio: DEPI/CPCIE/0027/2019
Asunto: Autorización de impresión
de tesis.

C. Carlos Antonio Olivas Pérez
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA.
PRESENTE

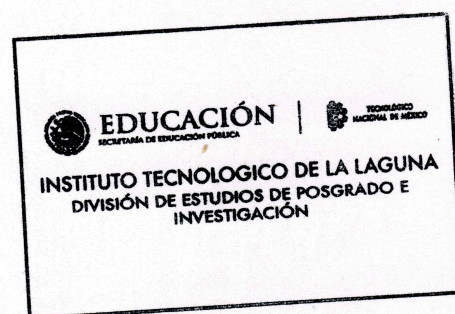
Después de haber sometido a revisión su trabajo de tesis titulado:

"Diseño y validación de un módulo de interfaz para transductores inteligentes a través del protocolo IEEE 21451"

Habiendo cumplido con todas las indicaciones que el jurado revisor de tesis hizo, se le comunica que se le concede la autorización con número de registro **RIITEC: (21)-TMCIE-2019**, para que proceda a la impresión del mismo.

ATENTAMENTE
EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN

DR. JOSÉ IRVING HERNÁNDEZ JACQUEZ
Jefe de la División de Estudios de Posgrado e Investigación
del Instituto Tecnológico de la Laguna





"2019, Año del Caudillo del Sur, Emiliano Zapata"

Torreón, Coah., 03/Diciembre/2019

DR. JOSÉ IRVING HERNÁNDEZ JACQUEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

Por medio de la presente, hacemos de su conocimiento que después de haber sometido a revisión el trabajo de tesis titulado:

"Diseño de un módulo e interfaz para transductores inteligentes a través del protocolo IEEE 21451"

Desarrollado por el **C. Carlos Antonio Olivas Pérez**, con número de control **M1713041** y habiendo cumplido con todas las correcciones que se le indicaron, estamos de acuerdo que se le conceda la autorización de la fecha de examen de grado para que proceda a la impresión de la misma.

ATENTAMENTE

EDUCACIÓN TECNOLÓGICA FUENTE DE INNOVACIÓN

Dr. Emmanuel Gómez Ramírez
Asesor/Director de Tesis

Dr. Hesner Coto Fuentes.
Coasesor Externo

Dr. Francisco Flores García
Comité Tutorial

Dr. José Irving Hernández Jacquez
Comité Tutorial



*Dedicado a Amelia Medrano Luna
Si llego a ser un gran hombre en mi vida te
lo deberé a ti, donde quiera que estés gracias.*

Agradecimientos

A mi familia, que me apoyaron incondicionalmente para lograr finalizar mis estudios. A mis padres, José Antonio Olivas Muruaga y María del Carmen Laura Pérez Medrano, por haberme forjado como la persona que soy en la actualidad; muchos logros en mi vida se los debo a ustedes, motivado por reglas y las oportunidades con las que me han permitido vivir.

A mi hermana, Ana Laura Olivas Pérez, quien siempre me dio su apoyo y cariño incondicional en mi día a día durante el transcurso de mi maestría.

A mis amigos, en especial a Francisco Eliezer Torrero Martínez, quienes han estado siempre presentes, y mucho más cuando los he necesitado, y que me animaron con su alegría y motivación a lograr este objetivo.

A mi asesor de proyecto, el Doctor Emmanuel Gómez Ramírez, y mi co-asesor, el Doctor Hesner Coto Fuentes, por su inmenso apoyo y dedicación a mi proyecto, quienes fueron apoyo fundamental en la elaboración de la presente tesis, impulsándome a lograr mis metas.

Y en general a todo el personal del Instituto Tecnológico de la Laguna, que me han ayudado durante todos estos años para alcanzar esta meta.

Resumen

Uno de los problemas que enfrentan la industria al fabricar un transductor es la compatibilidad para que estos puedan ser utilizado en las aplicaciones comunes del día a día. Además, la necesidad de poder saber de programación y el tiempo que conlleva la aplicación de la misma, provoca que los transductores se vuelvan un dolor de cabeza para usuarios inexpertos o para aquellos que requieren una solución rápida en la conexión de nuevos sensores o para el remplazo para actualización de los mismos. La IEEE a través del protocolo de IEEE 21451 propone la solución a estos problemas utilizando dispositivos *plug & play* (conecta y funciona), que son módulos con transductores que al conectarse otorgan al usuario la posibilidad de saber el tipo de componente con el que se está trabajando y las características de los transductores para poder hacer uso de él.

El modelo IEEE 21451 establece un conjunto de interfaces para hardware y software y un protocolo de red para la comunicación entre los distintos dispositivos, permitiendo separar el diseño de los transductores de la elección de las redes de comunicación. De esta manera el fabricante puede centrarse en su dispositivo de transducción, sin tener que preocuparse de su adaptación a diferentes redes,

lo que podría contribuir a mejorar la calidad de los transductores y a reducir sus costos. Este trabajo implementa el protocolo IEEE 21451 utilizando una PC como concentrador o NCAP, y un ARDUINO MEGA 2560 como STIM, por ser una placa de desarrollo basada en un microcontrolador re-programable bastante poderoso y que tiene 16 entradas analógicas que son útiles para conectar y controlar distintos transductores. Se muestra también el desarrollo de tres interfaces en Visual Basic, las cuales nos permiten enviar y recibir información del STIM, además de poder controlar la activación o lectura de los distintos canales disponibles de este.

En conclusión, la implementación del protocolo es eficiente y obtienen grandes ventajas de uso contra la forma usual de programar los transductores. Asimismo, este protocolo puede desarrollarse fácilmente en diferentes plataformas, sistemas, placas de computadoras o microcontroladores, lo que demuestra que existe una ventaja económica para los fabricantes, ya que permite centrarse únicamente en el diseño de los transductores dejando a un lado compatibilidad con las distintas formas de comunicación.

Índice general

Agradecimientos	II
Resumen	III
Índice de figuras	IX
Índice de tablas	XI
1. Introducción	1
1.1. Antecedentes del problema.	1
1.2. Planteamiento del problema	6
1.3. Objetivos	8
1.3.1. Objetivo General	8
1.3.2. Objetivos Específicos	9
1.4. Hipótesis	9
1.4.1. Preguntas de investigación.	9
2. Protocolo IEEE 21451	11

<i>ÍNDICE GENERAL</i>	VI
2.1. Clasificación de transductores	12
2.2. Señal y sistema	13
2.3. Acondicionamiento de señal	16
2.4. Protocolo IEEE 21451	18
2.4.1. La familia IEEE 21451	19
2.4.2. NCAP	22
2.4.3. STIM	23
2.4.4. TEDS	25
2.4.4.1. TEDS básico	27
2.4.4.2. Plantillas Estándar	27
2.4.5. Interfaz IEEE 21451.4 de modo mixto	31
2.4.5.1. TEDS Clase I	31
2.4.5.2. TEDS Clase II	32
2.4.5.3. TEDS virtual	32
2.4.6. Protocolo de Comunicación	33
A. Especificaciones de funciones de transductor in-	
teligente	33
2.4.7. Especificación de los TEDS	44
2.4.7.1. Bloque de datos de Meta-TEDS	46
2.4.7.2. Bloque de datos del TEDS de los canales	48
2.4.8. Interfaz Independiente del Transductor	49
3. Desarrollo de Interfaces	51

ÍNDICE GENERAL

VII

3.1. Desarrollo del sistema	51
3.2. Elección de Hardware	52
3.2.1. Interfaz para creación de UUID	52
3.2.1.1. Campo de localización	52
3.2.1.2. Campo de fabricante	55
3.2.1.3. Campo de año	55
3.2.1.4. Campo de tiempo	55
3.2.1.5. Pseudocódigo	56
3.2.1.6. Diagrama de flujo	57
3.2.1.7. Código de programación	58
3.2.2. Interfaz para creación de TEDS	67
3.2.2.1. Pseudocódigo	75
3.2.2.2. Diagrama de flujo	76
3.2.2.3. Código de programación	77
3.2.3. Interfaz para la visualización de datos, NCAP	95
3.2.3.1. Pseudocódigo	96
3.2.3.2. Diagrama de flujo	97
4. Pruebas experimentales y resultados del Protocolo IEEE 21451	98
4.1. Elección de Hardware	99
4.2. STIM	100
4.2.1. Escritura en la EEPROM del Arduino	100
4.2.2. Lectura desde el Arduino	103

<i>ÍNDICE GENERAL</i>	VIII
4.2.2.1. Pseudocódigo	104
4.2.3. Pruebas del módulo STIM y del NCAP	106
5. Conclusión y Recomendaciones	110
5.1. Conclusión	110
5.2. Recomendaciones	112
A. Tablas de datos para valores de los TEDS	114
Bibliografía	119

Índice de figuras

2.1. Esquema de un sensor.	13
2.2. Ejemplo de sensor y actuador.	14
2.3. Sistema de control de nivel de fluido. El tubo de visión y el ojo del observador forman el sensor.	15
2.4. Sensor complejo. El sensor tiene incorporado algunos transductores, además, e1, e2 y e3 son varios tipos de energía. A la salida se tiene un Sensor sencillo.	16
2.5. Posición de varios sensores en sistema adquirente de datos. El sensor 1 es un sensor sin contacto, el sensor 2 y 3 son sensores pasivos, el sensor 4 es activo, y el sensor 5 es interno en el sistema.	17
2.6. Arquitectura del conjunto de estándares IEEE 21451	19
2.7. Modelo de la conexión del transductor inteligente.	22
2.8. Contexto para las especificaciones de la interfaz del transductor.	23
2.9. Información contenida en la EEPROM del transductor.	26
2.10. Plantilla de termistor (número de identificación = 38).	30
2.11. Diseño de dirección.	34

<i>ÍNDICE DE FIGURAS</i>	X
2.12. Respuesta general del <i>STIM</i> al ser activado.	39
2.13. Protocolo general de transferencia de datos.	50
3.1. Interfaz para la creación de UUID.	53
3.2. Campo de localización, ubicación geográfica de la ciudad de Torreón, Coahuila, México.	54
3.3. Diagrama de flujo para la creación de UUID.	57
3.4. Plantilla de Acelerómetro/Transductor de fuerza (número de identificación = 25).	69
3.5. Plantilla de Acelerómetro/Transductor de fuerza (número de identificación = 25) continuación.	70
3.6. Interfaz para la creación de TEDS.	72
3.7. Plantilla de termistor (número de identificación = 38).	73
3.8. Diagrama de flujo para la creación de TEDS.	76
3.9. Interfaz del NCAP.	95
3.10. Diagrama de flujo para el NCAP.	97
4.1. Uso de la interfaz para la creación de TEDS.	101
4.2. Datos contenidos en la EEPROM.	102
4.3. Captura de datos para el UUID.	107
4.4. Captura de datos para el creador de TEDS.	108
4.5. Diagrama de conexión de potenciómetro.	108
4.6. Demostración del funcionamiento del NCAP.	109

Índice de tablas

2.1. Familia IEEE 1451	20
2.2. Familia IEEE 1451	21
2.3. Contenido del TEDS básico	27
2.4. Plantillas IEEE	29
2.5. Valor de bit para la dirección de la comunicación.	35
2.6. Funciones más comúnmente usada.	35
2.7. Estados STIM relacionados con la activación.	40
2.8. Comando de control estándar.	41
2.9. Bits del estatus estándar.	42
2.10. Bloque de la estructura de los datos del Meta-TEDS.	46
2.11. Bloque de la estructura de los datos del Meta-TEDS (continuación).	47
2.12. Bloque de la estructura de los datos del TEDS del canal.	48
2.13. Bloque de la estructura de los datos del TEDS del canal (continuación).	49
2.14. Líneas físicas.	50

ÍNDICE DE TABLAS

3.1. Tipo de datos usados en las plantillas estándar de la IEEE.	71
A.1. Valores para datos Chr5.	114
A.2. Números de identificación de los fabricantes.	115
A.3. Números de identificación de los fabricantes (continuación). . . .	116
A.4. Números de identificación de los fabricantes (continuación). . . .	117
A.5. Tipo de datos usados en las plantillas estándar de la IEEE.	118

Capítulo 1

Introducción

1.1. Antecedentes del problema.

La definición de sensor a menudo se describe como un dispositivo que recibe y responde a una señal o estímulo. Esta definición es burda y frecuentemente es confundido con un transductor. Este último es un dispositivo usado para convertir un tipo de energía en otro [7]. Un ejemplo de transductor es un altavoz que convierte un sistema eléctrico en un campo magnético variable y, posteriormente, en ondas acústicas. Cualquier sensor es un convertidor de energía. No importa lo que intente medir, siempre se ocupa de la transferencia de energía del objeto de medición al sensor, pero el proceso de detección es un caso particular de transferencia de información, y cualquier transmisión de información requiere transmisión de energía [7]. Los transductores pueden ser partes de sensores complejos. Por ejemplo, un sensor químico puede tener una parte que convierte la energía de una

1.1. ANTECEDENTES DEL PROBLEMA.

reacción química en calor (transductor) y otra parte, una termopila, que convierte el calor en una señal eléctrica. La combinación de estos dos forman un sensor químico, un dispositivo que produce una señal eléctrica en respuesta a una reacción química. Tenga en cuenta que, en el ejemplo anterior, un sensor químico es un sensor complejo; Está compuesto por un transductor y otro sensor (de calor). Esto sugiere que muchos sensores incorporan al menos un sensor de tipo directo y varios transductores.

Aunque pareciera que los transductores electrónicos tienen poco de ser desarrollados, la realidad es que estos existen desde 1874, cuando científicos franceses utilizaron la tecnología de onda corta en Mont Blanc, una montaña de 4,695 metros de altura, donde trataron de medir temperatura, velocidad del viento, altura de la nieve, etcétera. El uso de los transductores es vital para la vida cotidiana, ya que se encuentran en miles de dispositivos con los que interactuamos día con día. La utilización de transductores no solo es indispensable en procesos industriales y de manufactura donde está inmersa la robótica o la ingeniería experimental, también en sectores no productivos como son el ahorro de energético y el control ambiental [4].

El funcionamiento de sistemas de medición y control que se componen de transductores interconectados por una red de manera integrada aumenta la confiabilidad y desempeño del sistema [22]. En este contexto, la comprensión de la problemática actual del proceso de conexión tanto física como eléctrica del transductor a la red es un punto de vital importancia, pero debido a la ausencia de interfaces normalizadas para conexión de módulos con sensores en ambientes de

1.1. ANTECEDENTES DEL PROBLEMA.

red, se presentan complicaciones de diseño. El empleo de diferentes tecnologías o cambios en la aplicación, exigen el re-proyecto de las interfaces involucradas en el sistema.

La programación de los sistemas de medición ha generado grandes cambios, haciendo que los ingenieros e investigadores motivados a realizar desarrollos tecnológicos busquen nuevos métodos para mejorar la eficiencia de los procesos y de esta manera disminuir los tiempos de programación para utilización de los transductores y reducir el impacto que se ha tenido en el ambiente por la emisión de contaminantes al desarrollar piezas innecesarias.

A pesar de la creciente presencia de transductores inteligentes, los sensores analógicos tradicionales también están presentes en muchos sistemas modernos de medición y control. La conexión y el mantenimiento de dichos transductores, especialmente cuando se trata de sistemas complejos, es una tarea exigente y costosa. La experiencia muestra que aproximadamente el 20% del costo total de un sistema de adquisición de datos pertenece a la configuración del hardware y la calibración de los sensores. Además, también deben considerarse los problemas relacionados con los errores comunes que ocurren durante la entrada manual de parámetros importantes del sensor, como el rango de medición, la sensibilidad y los factores de escala, entre otros factores [15].

El surgimiento de la interfaz normalizada para transductores Inteligentes IEEE 21451 busca encauzar este tipo de problemas. Dicha norma fue creada con la finalidad de facilitar la conexión de transductores inteligentes a diferentes tipos de red [22].

1.1. ANTECEDENTES DEL PROBLEMA.

Conforme que los sensores y actuadores se vuelven más complejos, pueden brindar soporte para varios modos de operación. Algunas aplicaciones requieren además tolerancia a fallas y computación distribuida. Dichas funciones de alto nivel se pueden lograr agregando un microcontrolador incorporado al sensor / actuador clásico, convirtiéndolo en un transductor inteligente, lo que aumenta la capacidad de hacer frente a la complejidad a un precio justo.

Un transductor se considera que es inteligente cuando es combinado con una unidad de procesamiento y una interfaz de comunicación [21]. Los transductores inteligentes actuales son capaces de procesar datos muy complejos: identifican y comprueban con una exactitud cualquier dato que se necesite conocer. Las características de un transductor inteligente incluyen autoidentificación, autodiagnóstico, reconocimiento de ubicación, proceso de datos y fusión, y comunicación de red [23]. Esta funcionalidad generalmente simplifica la integración del transductor en aplicaciones en un entorno en red. El desarrollo de esta tecnología ha permitido aumentar de manera significativa la calidad y la eficiencia de los procesos de investigación, los procesos en el área médica y también ha tenido gran trascendencia en los procesos industriales. Las actividades de medición son mejor atendidas por transductores con identificación y datos integrados. La IEEE y las empresas involucradas desarrollaron sistemas *plug and play* para la comunidad de medición. Esto ayudó a la transición a sistemas más grandes con más transductores y al mismo tiempo introdujo un nuevo nivel de calidad en las mediciones [10].

Existen protocolos de comunicación serie estándar abierto que permite el intercambio bidireccional de datos de sensores y dispositivos que soportan esta tec-

1.1. ANTECEDENTES DEL PROBLEMA.

nología, como lo son los dispositivos *IO-Link*. Estos se conectan a un maestro y pueden transmitir estos datos a través de varias redes, buses de campo o buses de fondo, haciendo que los datos sean accesibles para una acción inmediata o un análisis a largo plazo a través de un sistema de información industrial. Cada transductor *IO-Link* tiene un archivo IODD (*IO Device Description*) que describe el dispositivo y sus capacidades [5]. Debe tenerse en cuenta que a pesar que la comunicación *IO-Link* es bidireccional, requiere un módulo I/O y un cable estándar de 4 pines, la alimentación del transductor con tres pines y la transmisión de la señal en un cuarto pin. Esto conlleva a que el fabricante desarrolle un módulo compatible con la tecnología *IO-Link* el cual va conectado directo al transductor que desarrolle, a diferencia del protocolo 21451 donde puede utilizar uno o dos cables para conectar el transductor, ya que el módulo para la comunicación es independiente del transductor a fabricar.

La arquitectura completa del protocolo 21451 de la IEEE básicamente se define en un Procesador De Aplicaciones Con Capacidad De Red (NCAP, *Network Capable Application Processor*) y un Módulo De Interfaz De Transductor Inteligente (STIM, *Smart Transducer Interface Module*). El NCAP consiste en un hardware y software cuya misión es proporcionar una pasarela entre los distintos STIMs y la red de usuario, además controla al STIM a través de un interfaz digital. El STIM consta de uno o varios transductores, electrónica de conversión y procesado de señal, además, unas hojas de datos electrónicas de los transductores (TEDS, *Transducer Electronic Data Sheets*) y un módulo de comunicación IEEE 21451.X referido al interfaz físico entre éste y el NCAP. Los TEDS contienen in-

1.2. PLANTEAMIENTO DEL PROBLEMA

formación tal como la identificación del transductor, su rango de medición o datos de calibración.

Entre las diferentes ventajas de IEEE 21451 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).

En este trabajo se planteó el diseño de un módulo económico que proporcione información precisa y de fácil acceso sin la necesidad de una programación previa para el uso entre distintos transductores, con ayuda del protocolo 21451 de la IEEE.

1.2. Planteamiento del problema

El principal problema a resolver es conocer acerca de los lineamientos del protocolo 21451 de la IEEE, que propone uso de dispositivos *plug & play* para transductores analógicos, además establece las bases para el desarrollo y el envío

1.2. PLANTEAMIENTO DEL PROBLEMA

de las hojas de datos electrónicas. Un problema presente en los transductores es la compatibilidad entre los distintos tipos existentes en el mercado y los protocolos de comunicación, así como la programación de los mismos, lo cual es fácilmente atacado gracias a este protocolo.

Se pretende implementar un dispositivo que incorpore un microcontrolador, ingeniería de conversión de señal y pines para de entradas analógicas, que además sea capaz de comunicarse con una computadora por un puerto USB, para diseñar un módulo que sea capaz de otorgar una pasarela de información entre los distintos transductores inteligentes conectados en los canales disponibles. Todo esto controlado por una interfaz de fácil acceso para el usuario, desarrollada en un software libre que proporcionará los comandos necesarios para manipular la información de los distintos canales del módulo.

El proyecto consiste en un módulo que se desarrollará en una tarjeta re-programable la cual podrá contener en su EEPROM los datos para localizar archivos guardado en la PC con la información para el uso de los transductores análogos comerciales ya existentes. La interfaz del NCAP podrá tener un fácil acceso a la EEPROM del STIM y a la información almacenada en los archivos de la PC de los distintos transductores, y por medio de una estructura *plug & play* el usuario podrá utilizar los transductores sin necesidad de tener una programación previa, obteniendo una respuesta inmediata a las necesidades que se desean cubrir con el transductor. Asimismo, la interfaz para la creación de TEDS se utilizará para permitir que los transductores análogos de nueva creación puedan conectarse con esta tecnología de *plug & play*.

1.3. OBJETIVOS

La interfaz estará desarrollada para escribir y leer información en el módulo a través del puerto serial y mostrará en pantalla la información de la hoja de datos electrónica del transductor en uso, además de los valores que se pueden estar sensando en tiempo real.

El uso de dispositivos *plug & play*, propuesta con el protocolo 21451 de la IEEE, con el fin de separar el diseño de los transductores de la elección de las redes de comunicación, es una fuerte herramienta para luchar con las problemáticas actuales de compatibilidad entre los distintos transductores existentes en el mercado y los protocolos de comunicación, así como la programación.

Teniendo una herramienta la cual permita a los fabricantes centrarse únicamente en el diseño de sus transductores sin la necesidad de enfocarse en la compatibilidad de estos con los distintos métodos de comunicación, permitirán que los transductores puedan tener un menor costo y una mayor calidad durante su fabricación, lo que podrá ser redituable también en los usuarios finales de los transductores y apoyando en su economía.

1.3. Objetivos

1.3.1. Objetivo General

Diseñar y validar un módulo e interfaz para transductores inteligentes implementando el protocolo 21451 de la IEEE.

1.4. HIPÓTESIS

1.3.2. Objetivos Específicos

1. Validar el protocolo 21452 de la IEEE, a través del desarrollo de software y hardware acoplados a este protocolo.
2. Desarrollar una interfaz de manejo sencillo con un software libre para la creación de hojas de datos electrónicas (TEDS).
3. Desarrollar una interfaz con un software para la creación y envío de los datos del Identificador Único Universal, UUID, hacia el STIM.
4. Desarrollar una interfaz con un software libre, NCAP, que permita al usuario la comunicación y control entre los distintos STIM disponibles.
5. Proponer un pseudocódigo para el desarrollo del STIM en una placa electrónica de hardware libre que incorpore un microcontrolador re-programable.

1.4. Hipótesis

Es posible desarrollar un módulo para sensores inteligentes que permita validar el protocolo 21451 de la IEEE utilizando un software libre para el desarrollo de las interfaces y una tarjeta con microcontrolador re-programable integrado para el módulo que controla los distintos transductores.

1.4.1. Preguntas de investigación.

- 1.- ¿Realmente es efectivo el protocolo 21451 propuesto en la IEEE?

1.4. HIPÓTESIS

- 2.- ¿Los datos obtenidos por el módulo son correctos a comparación de los métodos utilizados actualmente?
- 3.- ¿La información de los TEDS puede compartirse de manera más efectiva con ayuda de este protocolo?
- 4.- ¿Existe una ventaja económica para los fabricantes de sensores al desarrollar módulos de transductores inteligentes que utilicen TEDS?

Capítulo 2

Protocolo IEEE 21451

La industria de los transductores analógicos es una de las más grandes que existe en el mundo, sin embargo no están exentas de problemas, desde los que pueden ocasionar un pequeño disgusto, como un sensor de temperatura que falle dentro de un refrigerador haciendo que la comida se eche a perder; hasta aquellos que puedan traer consecuencias más grandes, como un sensor que reporte que el funcionamiento en el núcleo de un reactor es correcto cuando en realidad el reactor está a punto de entrar en fusión [8].

Un transductor es aquel dispositivo que es capaz de convertir una señal de una forma física en una señal correspondiente, pero de otra forma física. Existe 6 tipos de señales: mecánicas, térmicas, magnéticas, eléctricas, ópticas y moleculares, y cualquier dispositivo que convierta un tipo de energía en otro se le debería considerar un transductor. No obstante, en la práctica, se consideran transductores a aquellos que ofrecen una señal de salida eléctrica. Ello se debe al interés de este

2.1. CLASIFICACIÓN DE TRANSDUCTORES

tipo de señales en la mayoría de procesos de medida [4].

Sensor y transductor se emplean a veces como sinónimos, pero sensor sugiere un significado más extenso. Para otorgar una distinción a estos términos, la tendencia actual es utilizar el término sensor para designar a un transductor de entrada y actuador a un transductor de salida [4].

2.1. Clasificación de transductores

Como se comentó anteriormente existen dos clasificaciones principales de transductores: los sensores y los actuadores.

- Los sensores son dispositivos que, a partir de la energía del medio donde se mide, da una señal de salida legible por un sistema electrónico en función de la variable medida [4]. La Figura 2.1 muestra el esquema básico de un sensor.
- Los actuadores también reciben una entrada y generan una salida, pero funcionan en sentido opuesto a un sensor. Un actuador es un transductor que recibe información y produce una salida, consistente en alguna forma de energía física. Por ejemplo, cuando un celular vibra, lo hace por efecto de actuadores.

Un ejemplo claro de esta clasificación se muestra en la Figura 2.2 donde se tiene un sensor, el micrófono que es un transductor electroacústico, y un actuador, un altavoz que también es un transductor electroacústico. El sensor convierte

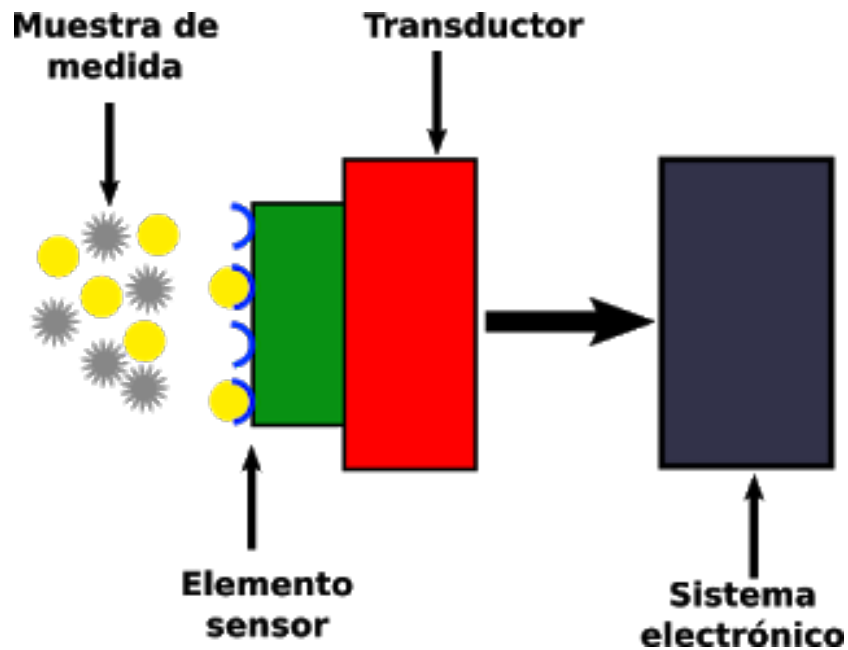


Figura 2.1: Esquema de un sensor.

la energía acústica (vibraciones sonoras) en energía eléctrica y con ayuda de un acondicionamiento de señal, esta señal eléctrica se convierte de nuevo en señal acústica que se emite por el parlante.

2.2. Señal y sistema

Un sensor es a menudo definido como un dispositivo que recibe y responde a una señal o a un estímulo. Pero esto no es del todo correcto. Utilicemos de ejemplo el sistema control de nivel mostrado en la Figura 2.3, el operador es capaz de cerrar o abrir la válvula observando el nivel que se encuentra en el tubo de visión. El operador debe obtener información sobre el nivel de fluido en el tanque



Figura 2.2: Ejemplo de sensor y actuador.

de manera oportuna para compensar las perturbaciones que tenga el tanque, si no esté probablemente se llegue a inundar o a vaciar, haciendo que se dañe el sistema. En este caso el sensor que percibe el nivel de fluido en el tanque consta de dos partes, el tubo de visión y el ojo del observador, ya que, en este caso en particular, ninguno de los dos funciona como un sensor de manera individual. Solamente la combinación de estos dos hace un sensor de propósito estrecho, que es selectivamente sensible al nivel de fluido [7].

Si el tubo está diseñado de manera correcta, este nos mostrará variaciones de nivel rápidamente, y se dice que la respuesta del sensor es rápida. Si el diámetro del tubo es pequeño, el nivel del tubo puede retrasarse con respecto al nivel del tanque. Esto en algunos casos puede ser despreciable, mientras que, en otros casos,

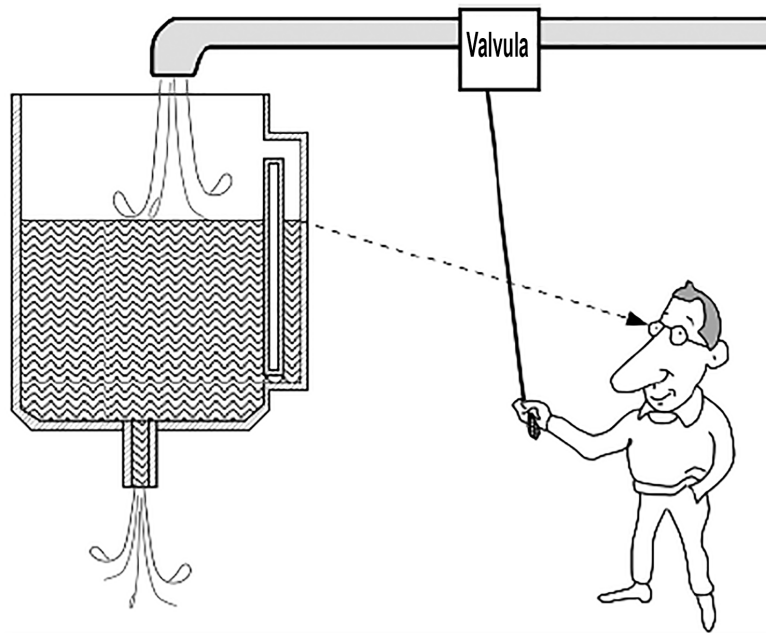


Figura 2.3: Sistema de control de nivel de fluido. El tubo de visión y el ojo del observador forman el sensor.

se requeriría un mejor diseño del tubo de visión. Por lo tanto, el rendimiento del sensor se debe evaluar sólo como parte de un sistema de adquisición de datos.

En resumen, hay dos tipos de sensores: sencillos y complejos. Un sensor sencillo convierte un estímulo en una señal eléctrica o modifica una señal eléctrica mediante el uso de un efecto físico apropiado, mientras que un sensor complejo además necesita uno o más transductores de energía antes de poder emplear un sensor directo para generar una salida eléctrica [7].

Un sensor no funciona por su propia cuenta; siempre es parte de un sistema que también incorpora otros detectores, acondicionadores de señal, procesadores de señal, dispositivos de memoria, registradores de datos y actuadores. Un sensor

2.3. ACONDICIONAMIENTO DE SEÑAL

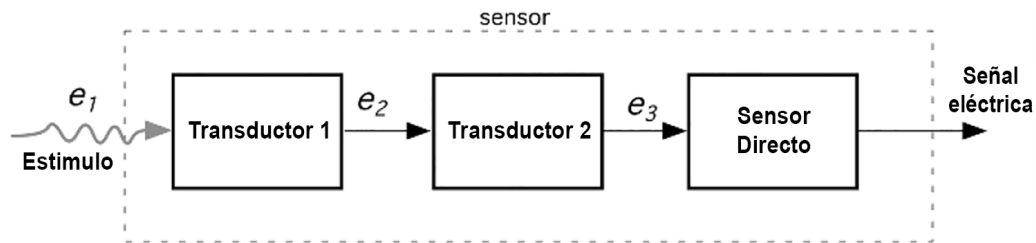


Figura 2.4: Sensor complejo. El sensor tiene incorporado algunos transductores, además, e_1 , e_2 y e_3 son varios tipos de energía. A la salida se tiene un Sensor sencillo.

siempre es parte de algún sistema adquiredor de datos y a menudo estos sistemas forman parte de sistemas de control más grandes con mecanismos de retroalimentación.

La Figura 2.5 muestra un sistema de adquisición de datos y un dispositivo de control conectado a distintos sensores; el sensor 2,3 y 4 se encuentran posicionados sobre o dentro del objeto que se está midiendo. El sensor 1 se encuentra fuera de este teniendo sus mediciones sin tener un contacto físico con el objeto, y se le llama sensor sin contacto. El sensor 5 tiene un diferente propósito, monitorear de manera interna las condiciones del sistema de adquisición de datos.

2.3. Acondicionamiento de señal

Los acondicionadores de señal o amplificadores, en un sentido más amplio, son los elementos del sistema de medida que ofrecen una señal apta que puede ser utilizada por un instrumento estándar, a partir de la salida de un sensor electrónico.

2.3. ACONDICIONAMIENTO DE SEÑAL

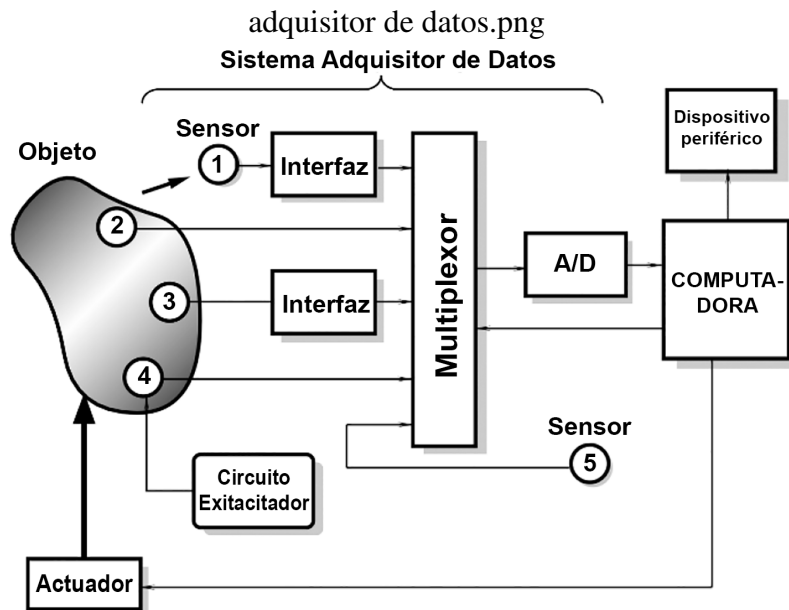


Figura 2.5: Posición de varios sensores en sistema adquirente de datos. El sensor 1 es un sensor sin contacto, el sensor 2 y 3 son sensores pasivos, el sensor 4 es activo, y el sensor 5 es interno en el sistema.

Entre las funciones que ofrecen los acondicionadores están las de amplificación, filtrado, adaptación de impedancias y modulación [4].

Considerando que en una etapa de tratamiento de señal es digital, y la salida del sensor es analógica, hará falta entre estas un convertidor A/D (Análogo-Digital). La salida analógica de un sensor está condicionada y amplificada y luego convertida a formato digital por un convertidor analógico a digital (A / D). La señal digitalizada del sensor puede ser procesada fácilmente por un microprocesador utilizando un algoritmo de aplicación [23]. El convertidor tiene una impedancia de entrada limitada y exige que la señal aplicada sea de continua o de frecuencia de variación lenta, y que su amplitud este entre unos límites determinados (nor-

2.4. PROTOCOLO IEEE 21451

malmente no mayores de 10v). Todas estas exigencias obligan a interponer un acondicionador de señal entre el sensor y el convertidor A/D.

2.4. Protocolo IEEE 21451

El protocolo IEEE 21451 establece un conjunto de interfaces para hardware y software con el fin de que los fabricantes puedan separar el diseño de los transductores de la elección de las redes de comunicación. Al permitir separar ambas partes, transductores y redes de comunicación, se podría contribuir a mejorar la calidad de los transductores y reducir sus precios.

La familia IEEE 21451 propone un conjunto de interfaces de comunicación para conectar transductores inteligentes a sistemas, alámbricos e inalámbricos, basados en microprocesadores, instrumentos y redes. La figura 2.6 muestra la arquitectura completa. El protocolo IEEE 1451 se puede dividir en dos partes principales: un procesador de aplicaciones con capacidad de red (o NCAP por sus siglas en inglés *Network Capable Application Processor*) y un módulo de interfaz de transductores inteligentes (o STIM por sus siglas en inglés *Smart Transducer Interface Module*) [24]. El NCAP realiza comunicaciones de red, comunicaciones con el STIM y conversión de datos u otras funciones de procesamiento. El STIM es un módulo que contiene una hoja de datos electrónica llamada *Transducer Electronic Data Sheet* (TEDS), la cual tiene información para identificar a los distintos transductores, su rango de medición o datos de calibración [14]. El STIM utiliza los TEDS para implementar la interfaz de los transductores, y cual-

2.4. PROTOCOLO IEEE 21451

quier conversión de señal o acondicionamiento de señal. La comunicación entre NCAP y STIM es a través de IEEE 21451.X. El estándar IEEE 21451 define diferentes tipos de comunicación del transductor, como punto a punto, distribuido *Multi drop Bus*, inalámbrico, CAN-bus y RFID. [24].

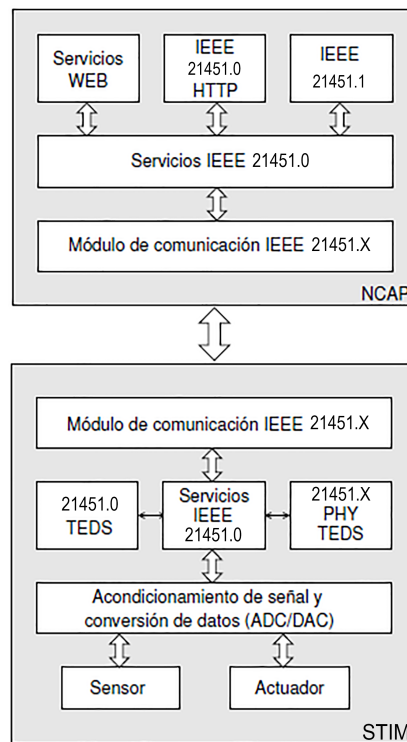


Figura 2.6: Arquitectura del conjunto de estándares IEEE 21451

2.4.1. La familia IEEE 21451

En el mes de septiembre de 1993, El Instituto Nacional de Estándares y Tecnología (NIST por sus siglas en inglés, *National Institute of Standards and Technology*) y el Instituto de Ingeniería Eléctrica y Electrónica (conocido por sus siglas

2.4. PROTOCOLO IEEE 21451

IEEE, en inglés *Institute of Electrical and Electronics Engineers*) promovieron una reunión para debatir acerca de las tecnologías existentes en el campo de transductores inteligentes, para lo que se crearon inicialmente cuatro grupos de trabajo, añadiéndose otros posteriormente [14].

Tabla 2.1: Familia IEEE 1451

Norma	Nombre	Descripción
21451.0	TEDS y Comandos Comunes	Esta norma define un conjunto común de comandos, funcionalidades y formas de TEDS para todos los miembros de la familia IEEE 1451.
21451.1	Interfaz NCAP-Computador	Define un modelo de información neutro, que se compone de un conjunto de jerárquico de clases que representan los diferentes bloques del NCAP.
21451.2	Interfaz punto a punto (TII)	Facilitar el soporte de múltiples redes y proporcionar una interfaz de comunicación común para la creación de transductores inteligentes en red.
21451.3	Buses Multi-Drop	El estándar define los interfaces eléctricos, TEDS, protocolo de identificación de canales, protocolos de sincronización y métodos de lectura y escritura para los TEDS y los datos de los transductores.

Todos los estándares de la serie IEEE 21451 están desarrollados para conectar con la interfaz de *plug and play* del sensor inteligente. Sin embargo, el sensor inteligente en red debe lograr no solo conectarse con la interfaz del transductor, sino también con la interfaz de red en forma *plug and play*. El *plug and play* universal es el estándar para el autorreconocimiento de dispositivos de red y las

2.4. PROTOCOLO IEEE 21451

Tabla 2.2: Familia IEEE 1451

Norma	Nombre	Descripción
21451.4	Interfaz Analógico	El objetivo es compatibilizar los transductores ya existentes en el mercado con el modelo IEEE 21451, centrándose principalmente en añadir la capacidad de almacenar TEDS a estos antiguos sensores.
21451.5	Wireless (WiFi, Zigbee, etc)	Es un estándar para la comunicación de sensores inalámbricos que define un conjunto de especificaciones para la comunicación entre el <i>Wireless Transducer Interface Module</i> .
21451.6	CAN Bus	Define un mapeado de los TEDS del 21451 a las entradas de dirección CANopen, así como mensajes de comunicación, procesado de datos, un parámetro de configuración e información de diagnóstico
21451.7	RFID	IEEE Define una interfaz y protocolo de comunicación entre transductores y sistemas RFID.

redes automáticas [6]. El estándar IEEE 21451.1 define un modelo de objeto común para los componentes abstractos de transductores inteligentes en red, junto con las especificaciones de interfaz para estos componentes. El modelo de objetos proporciona un marco extensible para el diseño e implementación de software para un sistema de medición y control distribuido [16].

2.4.2. NCAP

La interfaz especificadora de Hardware que se encuentra entre el sensor o actuador y el resto del Hardware, conocida como NCAP, es la parte física de las conexiones de los transductores inteligentes.

El hardware del NCAP consiste de un microprocesador y su circuitería de soporte, así como el hardware implementado para las conexiones y las interfaces de entrada/salida de los transductores. Todo esto se muestra en la figura 2.8. Los

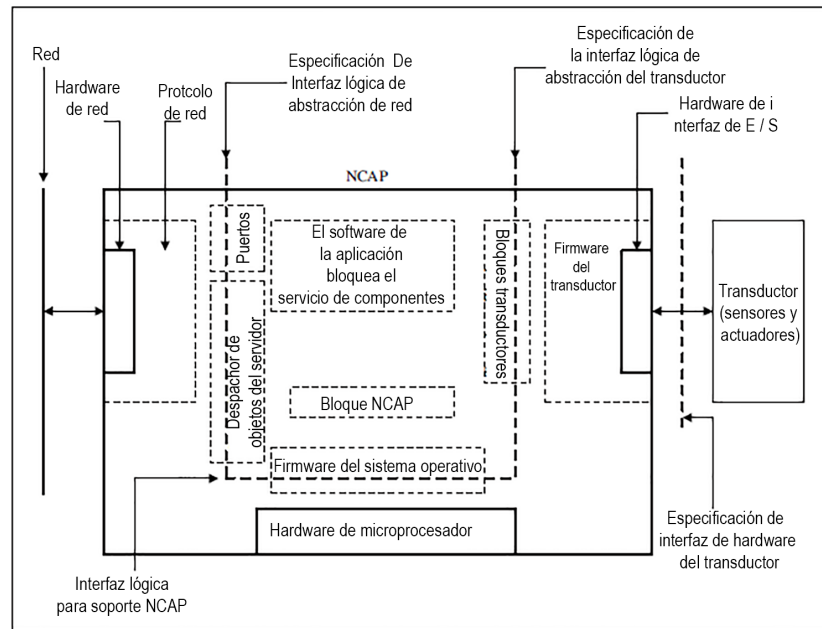


Figura 2.7: Modelo de la conexión del transductor inteligente.

componentes lógicos pueden ser en agrupados en aplicaciones y componentes de soporte. Los componentes de soporte son el sistema operativo, que provén una interfaz de la aplicación, los protocolos de comunicación y los firmwares de los componentes de los transductores [11].

2.4.3. STIM

Un STIM puede ser usado para sensar o controlar múltiples fenómenos físicos. Cada fenómeno debe estar asociado con un canal de transductor del STIM. Un canal puede ser un transductor virtual en el sentido que se comporta como un sensor o un actuador, aunque este no detecta ni cambia fuera del STIM [12].

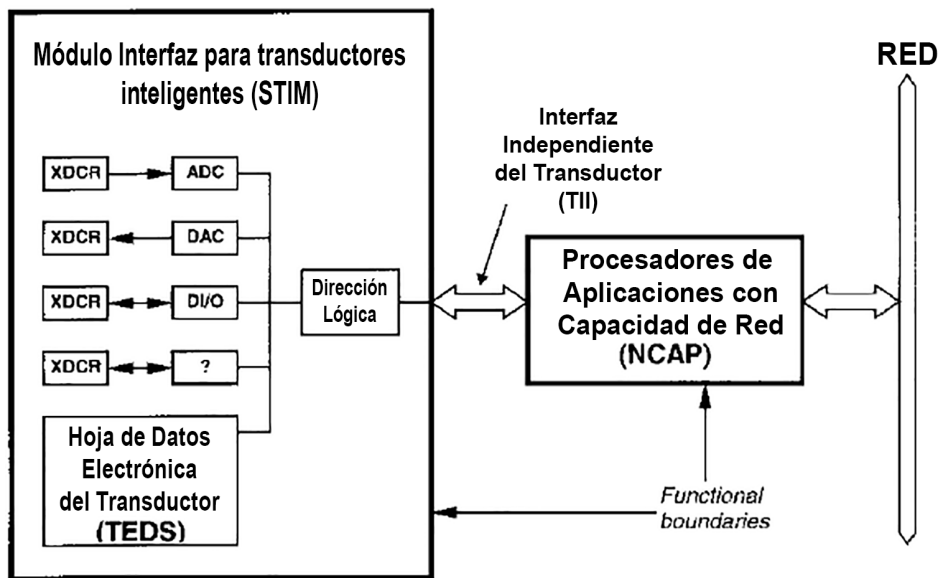


Figura 2.8: Contexto para las especificaciones de la interfaz del transductor.

Cada STIM debe implementar las siguientes funciones:

- Direccionalidad.
- Interfaz de transporte de datos.
- Meta-TEDS.

2.4. PROTOCOLO IEEE 21451

- Estatus Global.
- Control Global.
- Desencadenante.
- Capacidad de intercambio en caliente.
- Interrupción.
- Mascara de interrupción.

Asimismo, el STIM contiene el Identificador único universal (o UUID por sus siglas en inglés, *Universarl Unique Identification*) es un campo de identificación que está asociado con este y cuyo valor es único en el universo. Se utiliza el tiempo y la localización de la manufactura en un algoritmo para asegurar que sea único.

Cada canal del STIM debe implementar las siguientes funciones:

- TEDS del canal.
- Transporte de datos.
- Estatus.
- Control.

También cada canal puede implementar las siguientes funciones:

- TEDS de calibración.
- TEDS de identificación de calibración.

2.4. PROTOCOLO IEEE 21451

- TEDS de identificación de canal.
- TEDS de aplicaciones específicas de usuario final.
- TEDS de extensiones genéricas.
- Auto calibración.
- Auto prueba.

2.4.4. TEDS

El TEDS es un bloque de memoria asignado para describir las características de los transductores. Esta memoria puede residir en la memoria interna del procesador, pero con mayor frecuencia se encuentra en una memoria externa re-programable. La organización de datos en memoria se deja a la implementación particular de STIM [25].

La IEEE 21451.4 define a la estructura de los TEDS como múltiples secciones que juntas forman un completo TEDS. La primera sección son los TEDS básicos, los cuales comprimen la información esencial para poder identificar al transductor. Típicamente, enseguida del TEDS básicos, se define el TEDS estándar una estructura de datos que nos define las propiedades importantes de un transductor para su uso. Cada TEDS de canal tiene un TEDS de calibración opcional que traducirá los datos sin procesar del transductor a variables listas para ingeniería y puede ser byte, entero (16 bits), largo (32 bits), coma flotante o tipos de datos booleanos. El estándar no define específicamente la organización de estos datos [25]. Al final

2.4. PROTOCOLO IEEE 21451

se incluye un criterio del usuario para determinar el final de sección de los TEDS. Para indicar el inicio de cada sección se utilizan dos bits en los datos del TEDS [8].

Todos los TEDS usan el formato de tupla propuesto (tipo, longitud, valor o TLV) que consiste en el tipo de campo TEDS (1 byte), longitud (1 byte) y el valor (L bytes) [26].

En la figura 2.10 se muestra un ejemplo la información que puede estar contenida en el EEPROM del transductor inteligente.

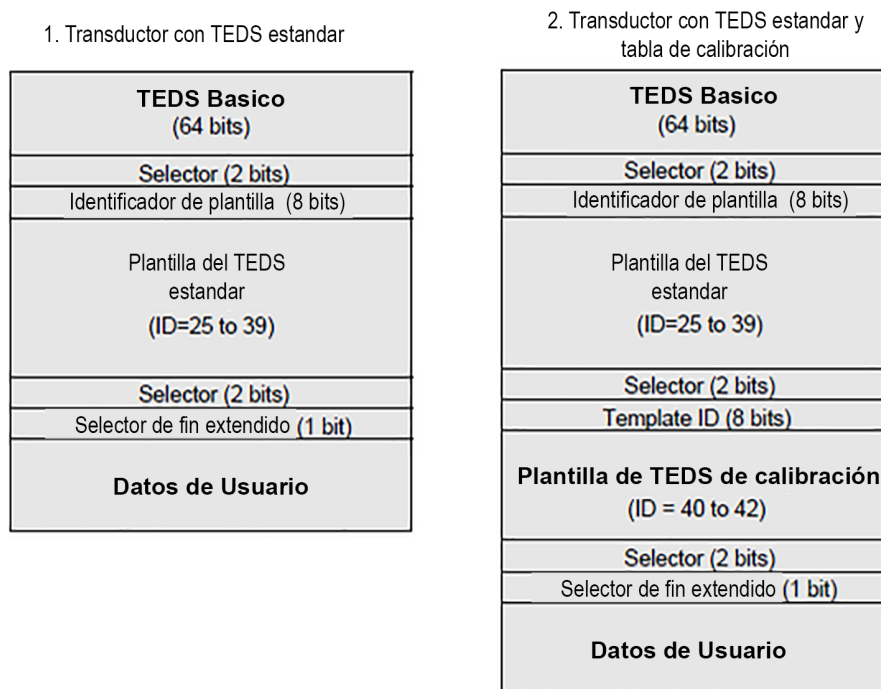


Figura 2.9: Información contenida en la EEPROM del transductor.

2.4. PROTOCOLO IEEE 21451

2.4.4.1. TEDS básico

La hoja de datos electrónicos del transductor (TEDS) en un transductor IEEE 21451.4 se almacena en una o más memorias direccionables independientemente llamadas nodos. Como mínimo, los TEDS deben contener un TEDS básico, que es una identificación única para el transductor. TEDS básico contiene un código de identificación del fabricante para el fabricante del transductor.

El TEDS básico está compuesto por 64 bits, incluida la ID del fabricante (14 bits), el número de modelo (15 bits), la letra de versión (código de caracteres de 5 bits), el número de versión (6 bits) y un número de serie de 24 bits para el transductor [9].

Tabla 2.3: Contenido del TEDS básico

	Longitud de Bit	Rango Disponible
Identificador de fabricante	14	17-16381
Numero de modelo	15	0-32767
Letra de la versión	5	A-Z(datos tipo Chr5)
Numero de la versión	6	0-63
Numero Serial	24	0-16777215

2.4.4.2. Plantillas Estándar

. El estándar define una colección de plantillas para las clases de transductores más comunes los cuales se en listan en la tabla 2.4, estas plantillas se enumeran con su propio número de identificación de cada plantilla. Para propósitos informativos, las plantillas se dividen en dos tipos: *Plantilla de tipo de transductor*,

2.4. PROTOCOLO IEEE 21451

defendidas para los tipos de transistores más comunes y las *plantillas de calibración*. Las plantillas de la 25 a la 39 y la plantilla 43 contienen propiedades específicas de cada tipo de transductor. Las plantillas 40, 41 y 42 son plantillas de calibración y se usan en conjunto con las plantillas de tipo de transistor [13].

Cada plantilla incluye la cantidad total de bits que requiere para poder ser implementada. La figura 2.10 muestra la plantilla correspondiente al Termistor, en la parte inferior de la tabla se muestra el resultado del cálculo de bits requeridos con el nombre *Total de Bits requeridos por el TEDS*. A menudo las plantillas muestran diferentes rangos de valores en la cantidad de bits que requieren, esto se debe a un valor de *SELECTCASE* en la plantilla. Como mínimo, la memoria del TEDS debe contener el TEDS Básico (64 bits), un selector de descripción al inicio de la plantilla (2 bits), el número de identificación de la plantilla (8 bits), el selector de descripción final de la plantilla (2 bits), y el selector extendido (1 bit) [13].

Tabla 2.4: Plantillas IEEE

Tipo	Identificador de plantilla	Nombre de la plantilla
Plantillas de tipo transductor	25	Acelerómetro / Transductor de fuerza con aplicación de corriente constante.
	26	Amplificador de carga (Incluye acelerómetro)
	27	Micrófono con preamplificador incorporado.
	28	Micrófono preamplificador con micr. o sistema adjunto
	29	Micrófono (capacitivo)
	30	Sensores de salida de voltaje de alto nivel
	31	Sensores de salida de bucle de corriente
	32	Sensor Resistivo
	33	Sensor de puente
	34	Sensor AC de Transformador diferencial de variación lineal/ rotacional (LVDT/RVDT)
	35	Medidor de tensión
	36	Par Termoeléctrico
	37	Detector resistivo de temperatura (RTDs)
	38	Termistor
	39	Divisor de tensión potenciométrico
Plantillas de calibración	40	Tabla de calibración
	41	Curva de calibración (polinomial)
	42	Tabla de respuesta a la frecuencia

Plantilla Termistor (ID = 38) resumen

Función	Descripción	Acceso	Bits
Identificación	Identificación de plantilla	—	8
Medición	Temperatura mínima	CAL	11
	Temperatura máxima	CAL	11
Salida de señal eléctrica	Tipo de señal eléctrica del transductor	ID	—
	Resistencia mínima de salida	CAL	18
	Resistencia máxima de salida	CAL	18
	Metodo de mapeo	ID	—
	Resistencia del termistor en 0°C	ID	20
	Coefficeinte A de Steinhart-Hart	ID	32
	Coefficeinte B de Steinhart-Hart	ID	32
	Coefficeinte C de Steinhart-Hart	ID	32
	Tiempo de respuesta del sensor	ID	6
Excitación o fuente de alimentación	Corriente de excitación nominal	CAL	8
	Corriente de excitación máxima	ID	8
	Constante de auto calentamiento	ID	5
Información de Calibración	Fecha de calibración	CAL	16
	Inicial de calibración	CAL	15
	Periodo de calibración	CAL	12
Diversos	ID. de localización de la medición	USR	11
Total de bits requeridos para el TEDS (rango)			263 bits

Figura 2.10: Plantilla de termistor (número de identificación = 38).

2.4. PROTOCOLO IEEE 21451

Por ejemplo, la plantilla del termistor requiere contener los siguientes bits:

<i>Checksum:</i>	8 Bits
<i>TEDS basico:</i>	64 Bits
<i>Selector de Descripción:</i>	2 bits
Total de Bits requeridos por el	
TEDS estándar (rango):	263 Bits
<i>Selector de Descripción (fin):</i>	2 Bits
<i>Selector extendido:</i>	1 Bits
Total de bits requeridos:	340 Bits

2.4.5. Interfaz IEEE 21451.4 de modo mixto

Un transductor inteligente proporciona una interfaz que conserva tanto una señal analógica tradicional y una interfaz digital que transfiere la información almacenada en la EEPROM. El estándar IEEE 21451.4 define dos clases de interfaz de modo mixto (MMI, por sus siglas en ingles), para permitir que el estándar se use con una selección de transductores tan amplia como sea posible [18], designadas como interfaces Clase I de dos cables y Clase II de múltiples cables.

2.4.5.1. TEDS Clase I

La interfaz de Clase I incluye diodos o conmutadores analógicos con los cuales es posible el multiplexado de la señal analógica con la información digital de

2.4. PROTOCOLO IEEE 21451

los TEDS a través del único par de cables existentes. El transductor incluye la EEPROM y la circuitería para la conmutación basada en corriente de polarización, que por lo general es solo una resistencia y uno o dos diodos.

2.4.5.2. TEDS Clase II

La interfaz Clase II utiliza conexiones diferentes para las señales analógicas y digitales de la interfaz de modo mixto. La entrada y salida analógica del transductor se deja sin modificaciones y el circuito digital para el TEDS se añade de forma paralela. Esto permite la implementación de transductores *plug & play* con prácticamente cualquier tipo de sensor o actuador [8].

2.4.5.3. TEDS virtual

La estructura de datos del TEDS puede describir cualquier sensor analógico, sin importar si el TEDS está incorporado o no en la EEPROM incorporada al sensor.

El estándar también establece el concepto de TEDS virtual. Permite descargar la información de miles de millones de sensores analógicos en lugar de tener un chip integrado.

Un archivo de TEDS virtual puede ser almacenado en una computadora local o en una base de datos de en una web accesible en lugar de ser contenida en la EEPROM. Esto permite que la inmensa instalación de sensores analógicos ya establecidos pueda aprovechar los beneficios del TEDS sin necesitar el EEPROM incorporado [17].

2.4.6. Protocolo de Comunicación

Un STIM es controlado por un módulo NCAP por medio de una interfaz digital dedicada. La interfaz no es una red. El NCAP media entre el STIM y la interfaz de red digital, y puede proporcionar inteligencia local.

El estándar 21451.2 define una interfaz digital para conectar transductores a microprocesadores. Además, describe a los TEDS y sus formas de datos, y define la interfaz eléctrica de las funciones lógicas de escritura y lectura para acceder a los TEDS [12].

La implementación del STIM debe seguir los 3 requerimientos siguientes:

1. El STIM soporta los requerimientos de ejecución de funciones especificados en el apartado A.
2. El STIM contiene los TEDS en el formato especificado en el apartado B.
3. La interfaz física del STIM líneas, protocolos y tiempos que están definidos en el apartado C.

A. Especificaciones de funciones de transductor inteligente

Para hacer emplear las funciones entre los distintos canales del *STIM* se utiliza el **direccionamiento**, el cual, usado en conjunto con la interfaz de transporte de datos, proporciona una dirección para especificar cuando un dato se debe leer o escribir, con cual función y en cual canal de *STIM*.

Una dirección completa debe contener de bytes de longitud y su estructura se muestra en la figura 2.11.

2.4. PROTOCOLO IEEE 21451

Dirección de Función Byte mas significativo						Dirección del Canal Byte menos significativo					
r/w	Código de la Función					Número del Canal					
msb					lsb	msb					lsb

Figura 2.11: Diseño de dirección.

Para conveniencia, el byte más significativo se le llama dirección de función y el menos significativo se le llama dirección de canal. La dirección de función depende de la dirección de canal, pero esta se transmite primero para hacer más sencilla la implementación.

Cada transductor en el *STIM* se le debe asignar un número de canal. Un *STIM* llegar a tener un máximo de 255 canales. El número de canales implementados debe ser determinado al leer el Meta-TEDS. Los canales implementados deben estar enumerados de forma consecutiva iniciando desde el uno.

La dirección de canal cero hace referencia al *CHANNEL_ZERO* del *STIM*. Cuando el *CHANNEL_ZERO* es usado, la función debe referir al *STIM* como un todo o como una colección de todos los canales. La palabra *global* o el prefijo *meta-* se utiliza en las funciones de este canal.

De acuerdo a la tabla 2.5, el bit más significativo de las direcciones de función especifica el rumbo de la comunicación de los datos en la interfaz. El resto de los bits de la dirección de función denotan la función seleccionada.

La tabla 2.6 muestra las funciones más comúnmente usada para cada dirección de función. La columna central es de para las funciones globales y la columna de la derecha son las funciones específicas de cada canal.

2.4. PROTOCOLO IEEE 21451

Tabla 2.5: Valor de bit para la dirección de la comunicación.

Valor	Dirección de la comunicación
0	Escribir en el STIM
1	Leer desde el STIM

Tabla 2.6: Funciones más comúnmente usada.

Dirección	Función Global	Función Canal
0	Escribir datos globales del transductor	Escribir datos del canal del transductor
1	Escribir datos globales del transductor	Escribir datos del canal del transductor
3	Escribir la dirección del canal activado	n/a
5	Escribir mascara global de interrupción estándar	Escribir mascara del canal de interrupción estándar
64	n/a	Escribir TEDS de calibración
65	n/a	Escribir número de identificación del TEDS de calibración
96	Escribir TEDS global de Aplicación específica del usuario final	Escribir TEDS del canal de Aplicación específica del usuario final
128	Leer información global de los transductores	Leer información del canal del transductor
130	Leer estatus estándar global	Leer estatus estándar del canal
132	Leer estatus auxiliar global	Leer estatus auxiliar del canal
160	Leer Meta-TEDS	Leer TEDS del canal
161	Leer identificación del Meta-TEDS	Leer identificación de canales TEDS
192	n/a	Leer TEDS de calibración
193	n/a	Leer identificación del TEDS de calibración
224	Leer TEDS global de Aplicación específica del usuario final	Leer TEDS del canal de Aplicación específica del usuario final

2.4. PROTOCOLO IEEE 21451

El formato de los datos que se envíen siempre deben de consistir en un número entero de bytes. Cuando los datos transportados involucran una representación numérica de múltiples bytes, el byte más significativo siempre se debe mandar primero. Cuando los bytes no representan un número múltiplo de 8 bits, se debe incluir un colchón de bits encima del bit más significativo hasta lograr un número total que sea múltiplo de 8. Cuando se deba enviar un número N-bytes que represente una fracción que no sea múltiplo de 8, el colchón debe incluirse al final del bit menos significativo hasta formar un número que sea múltiplo de 8.

La interfaz física debe transportar cada byte de los datos en la forma serial de bits, el bit más significativo primero.

Los medios de comunicación deben proveer en la interfaz física del NCAP una señal cuando el transporte de los datos está activado, y delimitar los marcos de los datos transportados. También debe proveerse de parte del STIM un *acknowledge* para avisar que está listo para el transporte de datos.

La función del transporte de datos interactúa con la función de disparo. La función de transporte de datos debe estar inactivo antes de que el disparo sea impongá.

El transporte de datos siempre tiene que comenzar con el NCAP enviando una dirección al *STIM*. La dirección completa especifica si el dato se va a escribir o leer desde el *STIM*, por cual canal y cual función está involucrada.

El tamaño de las estructuras de datos que se transportan se determinan a partir de una de las tres siguientes maneras:

- a) El tamaño del estatus, máscaras de interrupción, y de control de datos, y de los

2.4. PROTOCOLO IEEE 21451

disparos de la dirección del canal son especificado en el estándar.

- b) La longitud de los datos del transductor serán determinados por el campo de modelo del canal y los campos relacionados en el TEDS del canal.
- c) Todas las demás estructuras deben ser precedidas por un entero de 4 byte indicando la longitud de la estructura, excluyendo el campo de longitud de 4 byte.

El transporte de los datos se utiliza frecuentemente para leer información de los sensores y para escribir información en los actuadores. Los disparos de activación interactúan con el transporte de los datos.

Si se escribe un dato en un sensor este no tendrá ningún efecto.

Leer desde un sensor que no haya sido activado regresara el mismo dato que se leyó por última vez.

Leer un dato de un sensor después de haber abortado un ciclo de activación pueden producir resultados inesperados.

Leer un dato de un sensor amortiguado o de un sensor amortiguado de secuencia de datos después de inicializarse o después de abortar una secuencia de activación puede producir un resultado inesperado hasta que se realice el segundo disparo.

Leer información desde un actuador siempre regresara el último dato escrito en él.

Leer información desde un actuador después de inicializarse, pero antes de escribir algo en él, siempre regresara el estado inicial del actuador.

2.4. PROTOCOLO IEEE 21451

Las funciones del *CHANNEL_ZERO*, escribir datos globales en transductores y leer datos globales de los transductores, resulta en leer o escribir los datos de cada estructura de los canales implementados de forma concatenada, en orden, y comenzando con el primer canal. Esto permite que se tenga acceso a todos los transductores sin la necesidad de enviar las direcciones que esta asociada con cada canal individual. No existen delimitantes entre las estructuras de datos de cada canal. La información de cada canal debe ser procesada de acuerdo al modelo registrado en cada *TEDS* de los canales.

La transportación de información también puede ser usada para leer la hoja de datos contenidas en los *TEDS*, leer los estatus y escribir comandos de control en cada canal. Para el *CHANNEL_ZERO* los estatus de cada canal y sus comandos de control no se mandan de forma concatenada, en su lugar, se especifica un estatus global y las funciones de control globales. Siempre se aplicarán en el *STIM* como un todo.

La función de desencadenamiento provee medios para que el NCAP le envíe una señal al *STIM* para que una acción comience (señal de activación), y para que el *STIM* mande una señal en el tiempo que esta acción comienza (*acknowledgment* de activación). La activación debe aplicarse a un solo canal, o a todos los canales a la vez (activación global). Cada tipo de canal de transductor difiere de otro principalmente de la forma en que responde a la activación.

El diagrama de estados en la figura 2.12 ilustra el comportamiento de un sistema activado desde el punto de vista del *STIM*. La tabla 2.7 muestra los estados del *STIM* durante la activación [12].

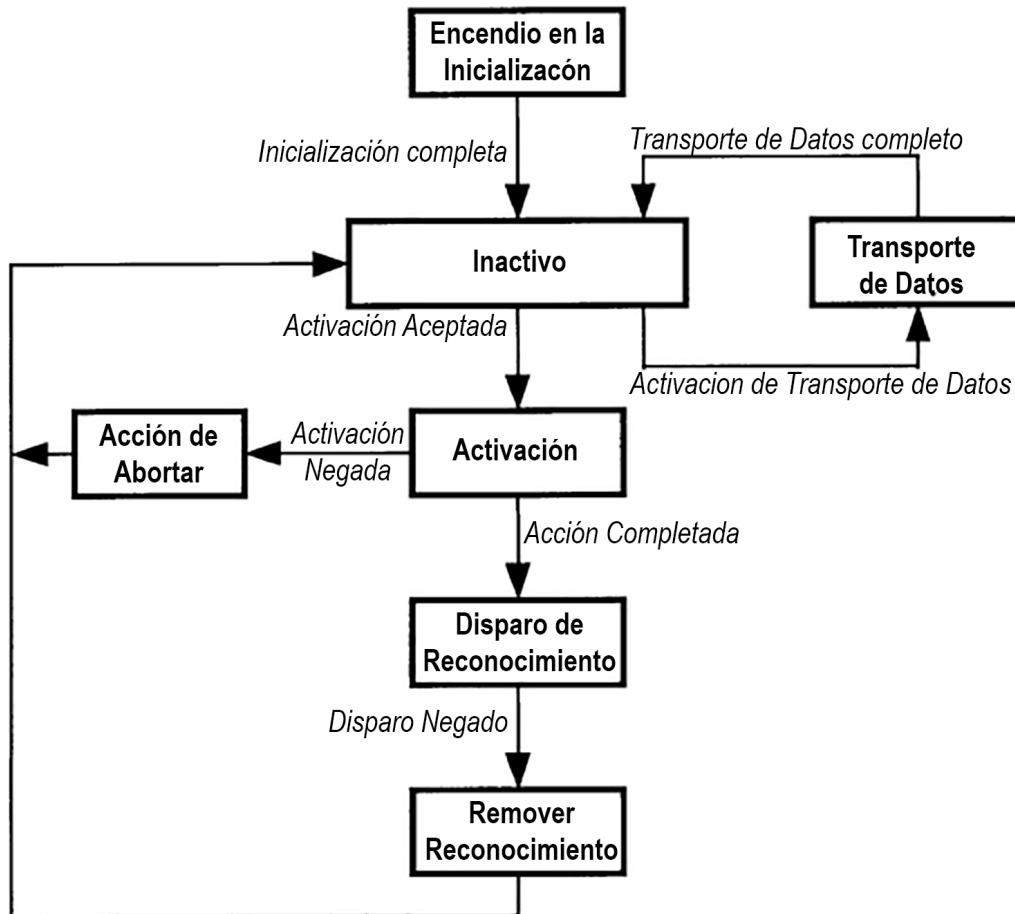


Figura 2.12: Respuesta general del *STIM* al ser activado.

2.4. PROTOCOLO IEEE 21451

Tabla 2.7: Estados STIM relacionados con la activación.

Estado	Definición
Inicialización de encendido	El STIM se está inicializando
Inactivo	El STIM está esperando que el activador o el transporte de datos estén activos
Activo	El STIM ha recibido la activación y el canal activado está listo para las acciones apropiadas
Abortar acción	El NCAP ha removido la activación prematuramente para abortar la acción en el STIM. Los datos resultantes del transductor o el estado del actuador son indefinidos. La máquina de estados procede directamente al estado inactivo
Reconocimiento de activación	El STIM ha reconocido la activación y está esperando por la negativa del NCAP
Remover reconocimiento	El NCAP ha negado la activación. El STIM remueve el reconocimiento de activación. La máquina de estados procede directamente al estado inactivo
Transporte de datos	El STIM está leyendo o escribiendo información, estatus, control o los campos de datos de los TEDS. La activación debe permanecer inactiva (negada) hasta que la transportación de datos este completa

Las funciones de control permiten enviar comando al *STIM* como un todo, o para cada canal del mismo, lo cual afecta a sus estados o sus operaciones. El acceso a las funciones de control se da a través de la dirección de función *Escribir comando de control del canal* para un canal específico o *Escribir comando de control global* para el *CHANNEL_ZERO*. Los comandos de control solo deben estar contenidos en 2 bytes.

Los comandos de control se definen en la tabla 2.8.

2.4. PROTOCOLO IEEE 21451

Tabla 2.8: Comando de control estándar.

Valor	Definición CHANNEL_ZERO	Definición canales individuales
0	Sin operación	Sin operación o
1	Reiniciar el STIM	Reiniciar el canal
2	Iniciar auto-prueba del STIM	Iniciar auto-prueba del canal
3	Calibrar todos los canales	Calibrar el canal
4	Iniciar a cero todos los canales	Iniciar a cero el canal
5	Habilitar todos los sensores de secuencia de eventos	Habilitar el sensor de secuencia de eventos
6	Deshabilitar todos los sensores de secuencia de eventos	Deshabilitar el sensor de secuencia de eventos
7	Poner todos los sensores de secuencia de eventos en modo de configuración	Poner el sensor de secuencia de eventos en modo de configuración
8	Reservado	Reservado
9	Habilitar todos los sensores de secuencia de datos o de secuencia de datos almacenados	Habilitar el sensor de secuencia de datos o de secuencia de datos almacenados
10	Deshabilitar todos los sensores de secuencia de datos o de secuencia de datos almacenados	Deshabilitar el sensor de secuencia de datos o de secuencia de datos almacenados
11-255	Reservado	Reservado
256-65 535	Abierto para la industria	Abierto para la industria

Las funciones de estatus permiten que el NCAP determine el estado del *STIM* como un todo o de forma individual cada canal. Cada bit en un registro específico del estatus representa la presencia o ausencia de una condición particular. La presencia de una condición se debe de representar con un uno en la apropiada posición del bit.

Las funciones de estatus también se utilizan en conjunto con las máscaras de interrupción y las interrupciones para indicar que el *STIM* está solicitando servi-

2.4. PROTOCOLO IEEE 21451

cio, y el porqué de la solicitud. Los registros de los estatus estándares y auxiliares son definidos a continuación y son implantándose para el *CHANNEL_ZERO* y para cada canal independiente en el *STIM*.

Tabla 2.9: Bits del estatus estándar.

Bit	Definición CHANNEL_ZERO	Definición canales individuales
msb	Abierto para la industria	Abierto para la industria
—	Abierto para la industria	Abierto para la industria
—	Abierto para la industria	Abierto para la industria
—	Abierto para la industria	Abierto para la industria
—	Reservado	Reservado
—	Reservado	Reservado
—	Reservado	Reservado
—	Bit operacional del STIM	Bit operacional del canal
—	OR de bits de error de hardware de los canales del STIM	Bit de error de hardware del canal
—	OR de bits de datos/eventos de todos los canales del STIM	Bit de datos/eventos del canal
—	OR de bits de datos o eventos perdidos de todos los canales del STIM	Bit de datos o eventos perdidos del canal
—	OR de bits de estatus auxiliares disponible de todos los canales del STIM	Bit de estatus auxiliares disponible del canal
—	Bit de comando invalido del STIM	Reservado
—	Bit de que se reinició el STIM	Bit de que se reinició el canal
—	Bit de reconocimiento de activación del STIM	Bit de reconocimiento de activación del canal
lsb	Bit de solicitud de servicio global	Bit de solicitud de servicio del canal

2.4. PROTOCOLO IEEE 21451

Se puede acceder a la lectura de los registros del estatus estándar a través de la dirección de la función *Leer estatus estándar del canal* para un canal específico o con la dirección de función *Leer estatus estándar global* para el *CHANNEL_ZERO*. Los bits del estatus estándar son mostrados en la tabla 2.9.

Para poder acceder a la lectura de los registros de los estatus auxiliares se utiliza la dirección de la función *Leer estatus auxiliar del canal* para un canal específico o con la dirección de función *Leer estatus auxiliar global* para el *CHANNEL_ZERO*. El estado devuelto será de 2 bytes de ancho. Todos los bits del estatus auxiliar son opcionales. El fabricante del STIM puede determinar a su criterio el uso de estos.

El STIM siempre contendrá tanto un registro de máscara de interrupción y un registro de máscara auxiliar. Ambas máscaras de interrupción tendrán una longitud de 2 bytes. Escribir un uno en cualquier posición de cualquier registro de las máscaras de interrupción permitirá establecer el bit de solicitud de servicio.

La posición de los registros de la máscara de interrupción estándar y de la máscara de interrupción auxiliar corresponden uno a uno a la posición de los bits del registro del estatus estándar y del registro de estatus auxiliar respectivamente. El valor localizado en el bit menos significativo de los registros de la máscara de interrupción estándar no es usada como correspondiente al bit de servicio. Ya que el bit de solicitud de servicio genera directamente la interrupción, no se puede enmascarar.

Se proporcionará una señal digital separada en la interfaz física para permitir que el STIM solicite servicio al NCAP. La señal de interrupción se usará en con-

2.4. PROTOCOLO IEEE 21451

junto con el registro de estatus y las máscaras de interrupciones indicar condiciones excepcionales en el STIM. El NCAP no requiere responder a una interrupción de forma inmediata.

La capacidad de *Hot-Swap* es necesaria porque el NCAP está conectado a una red y no puede apagarse para conectar o desconectar un STIM.

Se tiene que tener la posibilidad de crear o romper una conexión física entre el NCAP y el STIM mientras el NCAP sigue encendido, sin que exista daño alguno en cualquiera de las dos partes. La interfaz debe proveer los medios para detectar eventos de inserción y eventos de extracción.

Las lecturas y escrituras de datos están destinadas a acceder a estructuras de datos completas. Se permiten lecturas parciales de las estructuras de las hojas de datos de las funciones incluidas entre 160 y 255, y estas no deben ser tomadas como en un estado dañado. Durante el análisis deberá tomarse en cuenta cualquier dato incorrecto causado por esta lectura parcial.

Las lecturas parciales de las funciones entre 128 y 159, o las escrituras parciales en cualquier función, pueden dejar estas funciones en un estado corrupto.

2.4.7. Especificación de los TEDS

La cláusula define el formato de los bloques de datos TEDS de los cuales solo son obligatorios Meta-TEDS y el Canal TEDS. Todos los otros bloques de datos de los TEDS son opcionales. Todos los campos obligatorios de los TEDS siempre deben ser llenados, a menos que la longitud del campo sea cero.

2.4. PROTOCOLO IEEE 21451

Si un campo no es aplicable a la implementación, su valor será:

- Una cadena nula para el tipo de datos de cadena.
- Un NaN (*not a number*) para datos tipos reales de precisión simple y doble precisión.
- Un cero para enteros, enumeraciones y campos de longitud de datos.
- “Datos digitales” para tipos de datos de unidades físicas.

Los siguientes bloques de datos de los TEDS pueden tener un campo de longitud igual cero.

- Identificación de los Meta-TEDS.
- Identificación de los TEDS de los canales.
- Identificación de los TEDS de calibración.
- TEDS de calibración.
- TEDS de aplicaciones específicas de usuario final.
- TEDS de extensiones industriales.

Cuando la estructura de los datos se divide en bytes para hacer el envío de forma serial, que es correspondiente a un campo TEDS de múltiples bytes, el byte más significativo siempre se debe de transmitir primero. Un TEDS con un campo de longitud de $FFFFFFFF_{16}$ debe ser interpretado como una longitud de cero.

2.4. PROTOCOLO IEEE 21451

2.4.7.1. Bloque de datos de Meta-TEDS

La dirección de función número 160 aplicada al *CHANNEL_ZERO* dará acceso a este bloque de datos.

La función del Meta-TEDS es la de permitir a la interfaz tener la información necesaria para acceder a los canales, y la información en común de todos los canales. Los bytes del Meta-TEDS son constantes y solo son para lectura. En la tabla 2.10 y 2.11 se muestra la estructura de los datos.

Tabla 2.10: Bloque de la estructura de los datos del Meta-TEDS.

No. de campo	Descripción	Tipo	No. de bytes
Sub-bloque de datos relacionados con la constante de versión del TEDS			
1	Longitud del Meta-TEDS	U32L	4
2	Número de grupo de trabajo familiar estándar de la IEEE 1452	U8E	1
3	Número de versión del TEDS	U8E	1
Sub-bloque de datos relacionados con la identificación			
4	Identificador Global Único	UUID	10
Sub-bloque de datos relacionados con la identificación			
5	Llave de calibración para extensiones industriales de los TEDS del <i>CHANNEL_ZERO</i>	U8E	1
6	Llave de campo de datos no volátiles para extensiones industriales del <i>CHANNEL_ZERO</i>	U8E	1
7	Llave de extensiones industriales de los TEDS del <i>CHANNEL_ZERO</i>	U8E	1
8	Llave de aplicaciones específicas de usuario final de los TEDS del <i>CHANNEL_ZERO</i>	U8E	1
9	Número de canales implementados	U8C	1
10	Longitud del modelo de datos del canal en el peor de los casos	U8C	1
11	Las peores repeticiones de datos del canal	U16C	2

2.4. PROTOCOLO IEEE 21451

Tabla 2.11: Bloque de la estructura de los datos del Meta-TEDS (continuación).

No. de campo	Descripción	Tipo	No. de bytes
12	Longitud de TEDS escribible del CHANNEL_ZERO	U32C	4
Sub-bloque datos relacionados con el tiempo			
13	El peor tiempo de actualización del canal (t_{wu})	F32	4
14	Tiempo global de configuración de escritura (t_{gws})	F32	4
15	Tiempo global de configuración de lectura (t_{grs})	F32	4
16	Peor período de muestreo del canal (t_{wsp})	F32	4
17	Peor tiempo de calentamiento del canal	F32	4
18	Tiempo de respuesta de comando	F32	4
19	Tiempo de apretón de manos del STIM (t_{hs})	F32	4
20	Latencia de detección de fin de trama (t_{lat})	F32	4
21	Tiempo de espera del TEDS (t_{th})	F32	4
22	Tiempo de espera operacional (t_{oh})	F32	4
23	Velocidad de datos máxima	U32C	4
Sub-bloque datos relacionados con la agrupación de canales			
24	Longitud de sub-bloque de datos de agrupaciones de canales	U16L	2
25	Número de canales agrupados = G	U8C	1
Los campos del 26 al 28 se repiten G veces, una vez por grupo			
26	Tipo de grupo	U8E	1
27	Número de miembros del grupo = N	U8C	1
28	Lista de números de canales de miembros	Cadena de U8E	N
Sub-bloque de datos de integridad de datos			
29	Suma de comprobación para el Meta-TEDS	U16C	2

2.4. PROTOCOLO IEEE 21451

2.4.7.2. Bloque de datos del TEDS de los canales

La dirección de función número 160 aplicada a los canales del 1 al 255 proporcionan al acceso de estos datos. En la tabla 2.12 y 2.13 se muestra la estructura de los datos.

Tabla 2.12: Bloque de la estructura de los datos del TEDS del canal.

No. de campo	Descripción	Tipo	No. de bytes
Sub-bloque de datos relacionado con la estructura de datos			
1	Longitud del TEDS del canal	U32L	4
2	Llave de calibración	U8E	1
3	Llave de calibración para extensiones industriales de los TEDS del canal	U8E	1
4	Llave de campo de datos no volátiles para extensiones industriales del canal	U8E	1
5	Llave de extensiones industriales de los TEDS del canal	U8E	1
6	Llave de aplicaciones específicas de usuario final de los TEDS del canal	U8E	1
7	de TEDS escribibles del canal	U32C	4
Sub-bloque de datos relacionados con el transductor			
8	Llave de tipo de canal	U8E	1
9	Unidades físicas	UNITS	10
10	Límite de rango inferior	F32	4
11	Límite de rango superior	F32	4
12	Peor caso de incertidumbre	F32	4
13	Llave de auto-prueba	U8E	1
Sub-bloque de datos relacionados con la conversión de datos			
14	Modelo de datos del canal	U8E	1
15	Longitud del modelo de datos del canal	U8C	1
16	Bits significativos del modelo de canal	U16C	2
17	Repeticiones de datos del canal	U16C	2

2.4. PROTOCOLO IEEE 21451

Tabla 2.13: Bloque de la estructura de los datos del TEDS del canal (continuación).

No. de campo	Descripción	Tipo	No. de bytes
18	Origen de la serie	F32	4
19	Incremento de la serie	F32	4
20	Unidades de la serie	UNITS	10
Sub-bloque de datos relacionado con la sincronización			
21	Tiempo de actualización del canal (t_u)	F32	4
22	Tiempo de configuración de escritura del canal (t_{ws})	F32	4
23	Tiempo de configuración de lectura del canal (t_{rs})	F32	4
24	Período de muestreo del canal (t_{sp})	F32	4
25	Tiempo de calentamiento del canal	F32	4
26	Tiempo de espera agregado del canal (t_{ch})	F32	4
27	Correcciones de sincronización	F32	4
28	Precisión de disparo	F32	4
Campo de opciones de secuencia de eventos			
29	Opciones de secuencias de eventos	U8E	1
Sub-bloque de datos de integridad de datos			
30	Suma de comprobación para el TEDS del canal	U16C	2

2.4.8. Interfaz Independiente del Transductor

Los protocolos, el tiempo, y las especificaciones eléctricas son definidas para asegurar un transporte de datos entre las diversas combinaciones de STIMs y el NCAPs.

Las conexiones de líneas de la interfaz de los transductores del STIM y del NCAP se dividen en cuatro grupos. La tabla 2.14 muestra la lista de los grupos, el nombre completo y la abreviación estándar para cada línea.

El protocolo describe la implementación de la función de activación y la fun-

2.4. PROTOCOLO IEEE 21451

Tabla 2.14: Líneas físicas.

Grupo	Líneas	Abreviatura
Dato	DATA_OUT	DOUT
	DATA_IN	DIN
	DATA_CLOCK	DCLK
	N_IO_ENABLE	NIOE
Activación	N_TRIGGER	NTRIG
Soporte	POWER	POWER
	COMMON	COMMON
	N_ACKNOWLEDGE	NACK
	N_STIM_DETECT	NSDET
Interrupción	N_IO_INTERRUPT	NINT

ción de transporte de datos utilizando la implementación física de la Interfaz Independiente del Transductor (TII, *Transducer Independent Interface*). Tanto el NCAP como el STIM participan en cada protocolo y tienen roles ya definidos. Algunos protocolos están jerárquicamente definidos.

Los protocolos de más alto nivel son la lectura, escritura y la activación, en la figura 2.13 se ilustran las secuencias para la lectura y escritura entre el NCAP y el STIM.

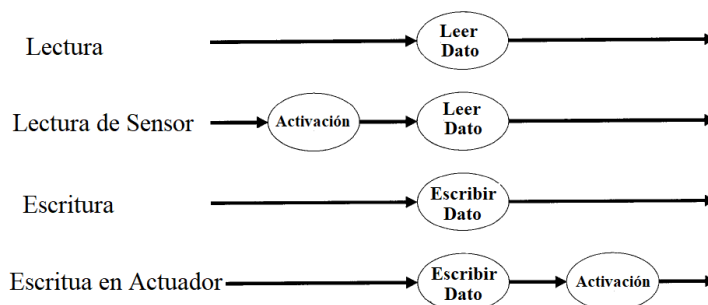


Figura 2.13: Protocolo general de transferencia de datos.

Capítulo 3

Desarrollo de Interfaces

3.1. Desarrollo del sistema

El proyecto se basa en un sistema que adapte un transductor genérico al protocolo 21451 de la IEEE para transductores inteligentes, a través de la norma 21451.4, cuyo objetivo es añadir la capacidad de utilizar TEDS virtuales para el reconocimiento y uso de estos transductores.

El sistema se adaptó a una interfaz TEDS clase II, para aprovechar las características de las dos líneas de comunicación (analógicas y digitales); se desarrollaron 3 interfaces en *Visual Basic* (La primera para la creación de UUID, la segunda para la creación de TEDS y la tercera el NCAP), las cuales son idealmente compatible con una plataforma de hardware libre que pueda contener en su estructura una EEPROM para la escritura y la lectura de los TEDS para hacerlo funcionar como STIM.

3.2. Elección de Hardware

Las interfaces se desarrollaron en el lenguaje de programación *C* con el programa *Visual Studio 2017*, que es un conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento, que utiliza el entorno de desarrollo integrado (IDE) y habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes, permitiendo el desarrollo de aplicaciones de escritorio y aplicaciones móviles [19]. Además, las funciones que se utilizan para la lectura y escritura del NCAP y el STIM son sencillas y no requiere softwares más especializados.

3.2.1. Interfaz para creación de UUID

El tamaño del campo del UUID es de 10 bytes y consta de 4 subcampos (en orden del más significativo y al menos significativo: localización, fabricante, año y tiempo). La figura 3.1 muestra la interfaz desarrollada en *Visual Basic*. Está dividida en los diferentes campos requeridos para la creación del algoritmo del UUID. A continuación, se define cada campo de manera individual.

3.2.1.1. Campo de localización

El valor de este campo siempre será escogido por el fabricante del STIM para identificar una ubicación particular en la tierra. Este valor puede representar la localización actual del fabricante del STIM. Los fabricantes pueden usar diferentes valores para este campo, siempre y cuando todos los valores sigan los siguientes

3.2. ELECCIÓN DE HARDWARE

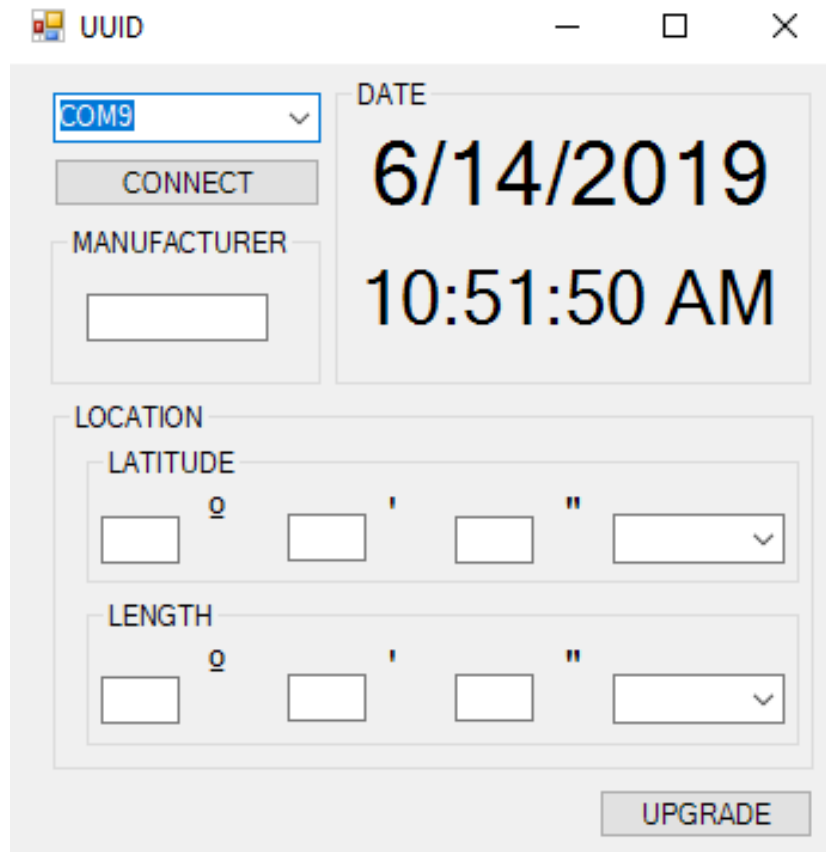


Figura 3.1: Interfaz para la creación de UUID.

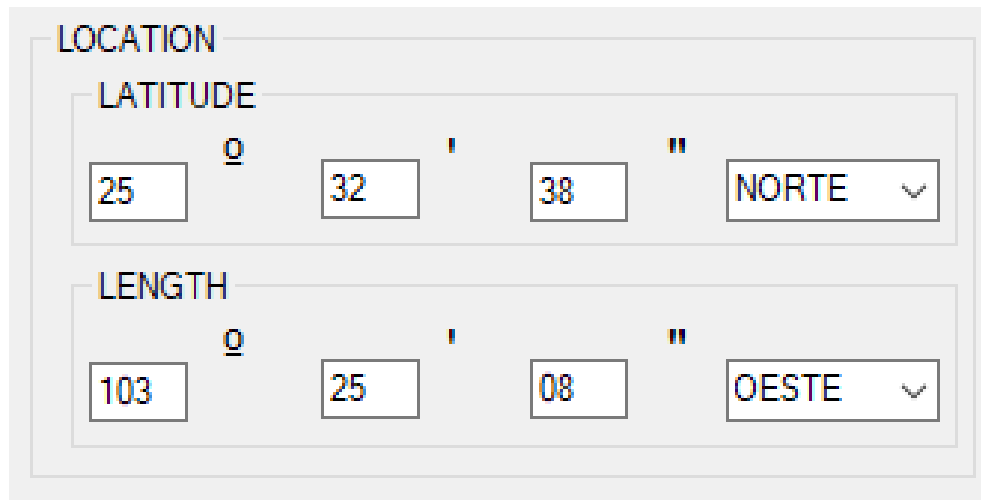
requerimientos.

- El campo de localización debe estar representado por 42 bits.
- El bit más significativo indica si es latitud Norte (1) o Sur (0).
- Los siguientes 20 msbs del campo representan la latitud de la localización en un numero entero de arcosegundos.
- El siguiente bit más significativo indica la latitud Este (1) u Oeste (0).

3.2. ELECCIÓN DE HARDWARE

- Los últimos 20 bits representan la latitud de la localización en número entero de arcosegundos.

La figura 3.2 muestra un ejemplo del campo de localización correspondiente a la ciudad de Torreón, Coahuila.



The image shows a user interface for entering location data. It is titled "LOCATION" and contains two main sections: "LATITUDE" and "LENGTH".

The "LATITUDE" section has three input boxes containing the values "25", "32", and "38". Above these boxes are the characters "°", "'", and "" respectively. To the right of these boxes is a dropdown menu with the text "NORTE" and a downward arrow.

The "LENGTH" section has three input boxes containing the values "103", "25", and "08". Above these boxes are the characters "°", "'", and "" respectively. To the right of these boxes is a dropdown menu with the text "OESTE" and a downward arrow.

Figura 3.2: Campo de localización, ubicación geográfica de la ciudad de Torreón, Coahuila, México.

(nota: un arcosegundo equivale a 30 m. El máximo rango representado en un campo de 20 bits es de 1, 048, 575 arcosegundos o 0-291 grados, lo cual es suficiente para representar latitud y longitud de la superficie de la tierra. Las magnitudes de latitud mayores a 90 grados y las magnitudes de longitud mayores de 180 grados están reservadas)

3.2. ELECCIÓN DE HARDWARE

3.2.1.2. Campo de fabricante

El valor de este campo es seleccionado por el fabricante del STIM. Cuando más de un fabricante tengan la misma la ubicación representada con el campo de localización, creando una interferencia, pueden negociar los valores de los campos de fabricante para resolver cualquier interferencia. Su tamaño es de 4 bits.

3.2.1.3. Campo de año

El valor de este campo representa al valor del año que transcurre. La representación del campo del año debe ser con un entero de 12 bits. El rango de años que cubre este campo es desde el 0 y hasta el 4095 D.C.

3.2.1.4. Campo de tiempo

La representación del campo de tiempo se da en un entero de 22 bits. El rango que abarca es de 0 hasta 4, 194,303 segundos. Existen aproximadamente 31, 536, 000 segundos por año, comenzando desde el primero de enero a las 00:00:00 horas.

Cuando es necesario conocer el valor de este campo, la interpretación son números enteros en intervalos de 10 segundos. En este caso los valores de tiempo mayores a un año son reservados [12].

3.2. ELECCIÓN DE HARDWARE

3.2.1.5. Pseudocódigo

El siguiente pseudocódigo representa a las funciones que se llevan a cabo durante la ejecución del programa para la creación del UUID.

- Inicio de programa.
- Selección de puerto a utilizar.
- Presionar botón *UPGRADE*.
- Iniciar función *LOCATION_FIELD*.
- Iniciar función *MANUFACTURER_FIELD*.
- Iniciar función *YEAR_FIELD*.
- Iniciar función *TIME_FIELD*.
- Concatenar los valores en el siguiente orden, Localización + Fabricante + Año + Tiempo, obteniendo un numero binario de 80 bits.
- Enviar el valor representativo de la dirección de la Función.
- Enviar el valor representativo de la dirección del Canal.
- Enviar el dato representativo del UUID.
- Esperar bit de acknowledge.
- Fin del programa.

3.2. ELECCIÓN DE HARDWARE

3.2.1.6. Diagrama de flujo

La figura 3.3 muestra un diagrama de flujo que representa la secuencia de las operaciones durante la ejecución del programa para la creación del UUID.

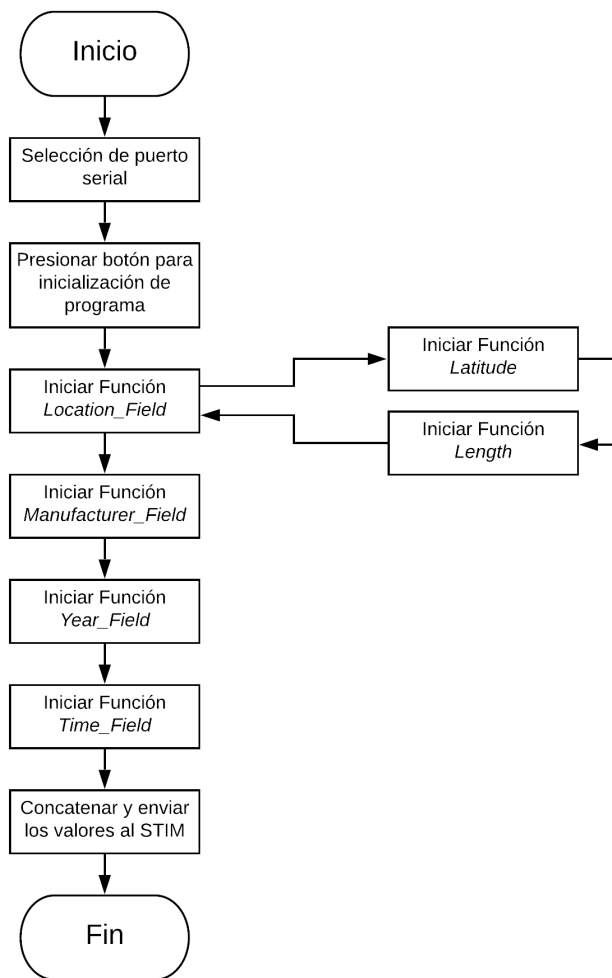


Figura 3.3: Diagrama de flujo para la creación de UUID.

3.2. ELECCIÓN DE HARDWARE

3.2.1.7. Código de programación

Esta sección muestra parte del código que se utilizó para la interfaz creadora de UUID.

Función *Location_Field*:

Función *LATITUDE*:

```
\\* CONVERTIR EL VALOR DE LATITUD EN BINARIO *//  
    private void LATITUDE()  
    {  
        \\* OBTIENE LOS VALORES DE LOS CAMPOS GRADOS, *//  
        \\* MINUTOS Y SEGUNDOS Y LOS CONVIERTE EN ENTEROS *//  
        GRADE = lat_field_grade.Text.Length > 0 ?  
            int.Parse(lat_field_grade.Text) : 0;  
  
        MINUTE = lat_field_min.Text.Length > 0 ?  
            int.Parse(lat_field_min.Text) : 0;  
  
        SECOND = lat_field_sec.Text.Length > 0 ?  
            int.Parse(lat_field_sec.Text) : 0;  
  
        SECOND = (GRADE * 3600) + (MINUTE * 60)  
            + SECOND;
```

3.2. ELECCIÓN DE HARDWARE

```
\\*OBTIENE EL VALOR AL PUNTO CARDINAL*//  
    POINT = N_S.SelectedIndex;  
  
    LATITUDE_FIELD = "";  
  
\\*OBTIENE EL VALOR BINARIO DE LA LATITUD*//  
    while (SECOND > 0)  
    {  
  
        LATITUDE_FIELD = SECOND % 2  
        + LATITUDE_FIELD;  
  
        SECOND = SECOND / 2;  
    }  
  
    DATA_LENGTH = LATITUDE_FIELD.Length;  
  
    RES = 20-DATA_LENGTH;  
  
    ADJ = LATITUDE_FIELD;
```

3.2. ELECCIÓN DE HARDWARE

```
\\\* REALIZA UN AJUSTE PARA GENERAR UNA MATRIZ *///  
\\\* DE 21 BITS *///  
ADJUSTMENT();  
  
LATITUDE_FIELD = POINT + ADJ;  
}
```

Función *LENGTH*:

```
\\\* CONVERTIR EL VALOR DE LONGITUD EN BINARIO *///  
private void LENGTH()  
{  
\\\* OBTIENE LOS VALORES DE LOS CAMPOS GRADOS, *///  
\\\* MINUTOS Y SEGUNDOS Y LOS CONVIERTE EN ENTEROS *///  
GRADE = len_field_grade.Text.Length > 0 ?  
int.Parse(man_field.Text) : 0;  
  
MINUTE = len_field_min.Text.Length > 0 ?  
int.Parse(man_field.Text) : 0;  
  
SECOND = len_field_sec.Text.Length > 0 ?  
int.Parse(man_field.Text) : 0;
```

3.2. ELECCIÓN DE HARDWARE

```
SECOND = (GRADE * 3600) + (MINUTE * 60)
+ SECOND;
```

```
\\*OBTIENE EL VALOR AL PUNTO CARDINAL*//
```

```
POINT = E_W.SelectedIndex;
```

```
LENGTH_FIELD = "";
```

```
\\*OBTIENE EL VALOR BINARIO DE LA LONGITUD*//
```

```
while (SECOND > 0)
{
    LENGTH_FIELD = SECOND % 2
    + LENGTH_FIELD;

    SECOND = SECOND / 2;
}
```

```
DATA_LENGTH = LENGTH_FIELD.Length;
```

```
RES = 20 - DATA_LENGTH;
```

```
ADJ = LENGTH_FIELD;
```

3.2. ELECCIÓN DE HARDWARE

```
\\* REALIZA UN AJUSTE PARA GENERAR UNA MATRIZ *///  
    \\* DE 21 BITS *///  
    ADJUSTMENT();  
  
    LENGTH_FIELD = POINT + ADJ;  
}
```

Función *Manufacturer_Field*

```
private void MANUFACTURER_FIELD()  
{  
    \\* OBTIENE LOS VALORES DEL CAMPO DE MANUFACTURA *///  
    int MANUFACTURER = man_field.Text.Length > 0 ?  
        int.Parse(man_field.Text) : 0;  
  
    \\* OBTIENE EL VALOR BINARIO DE MANUFACTURA *///  
    while (MANUFACTURER > 0)  
    {  
        MANUFACTURER_F = MANUFACTURER % 2  
        + MANUFACTURER_F;  
  
        MANUFACTURER = MANUFACTURER / 2;  
    }
```

3.2. ELECCIÓN DE HARDWARE

```
DATA_LENGTH = MANUFACTURER_F.Length;

RES = 4 - DATA_LENGTH;

ADJ = MANUFACTURER_F;

\\* REALIZA UN AJUSTE PARA GENERAR UNA MATRIZ *//
    \\* DE 4 BITS *//
    ADJUSTMENT();

    MANUFACTURER_F = ADJ;
}
```

Función *Year_Field*

```
private void YEAR_FIELD()
{
    \\* OBTIENE LOS VALORES DEL A O EN CURSO *//
    YEAR = d.Year;

    YEAR_F = "";

    \\* OBTIENE EL VALOR BINARIO DEL A O EN CURSO *//
    while (YEAR > 0)
```

3.2. ELECCIÓN DE HARDWARE

```
{
    YEAR_F = YEAR % 2
    + YEAR_F;

    YEAR = YEAR / 2;
}

DATA_LENGTH = YEAR_F.Length;

RES = 12 - DATA_LENGTH;

ADJ =YEAR_F;

\\* REALIZA UN AJUSTE PARA GENERAR UNA MATRIZ *///
    \\* DE 12 BITS *///
ADJUSTMENT();

YEAR_F =ADJ;
}
```

3.2. ELECCIÓN DE HARDWARE

Función *Time_Field*

```
private void TIME_FIELD()
{
  \\*OBTIENE LOS VALORES DEL D A EN CURSO*///
  DAY = d.DayOfYear;

  HOUR = d.Hour;

  MINUTE = d.Minute;

  SECOND = d.Second;

  \\*CONVIERTE LOS VALORES EN INTERVALOS DE 10 SEGUNDOS*///
  SECOND = ((DAY*86400)+(HOUR * 3600) + (MINUTE * 60)
  + (SECOND))/10;

  TIME_F = "";

  \\*OBTIENE EL VALOR BINARIO DEL D A EN CURSO*///
  while (SECOND > 0)
  {
    TIME_F = SECOND % 2 + TIME_F;
```


3.2. ELECCIÓN DE HARDWARE

```
        SECOND = SECOND / 2;
    }

    DATA_LENGTH = TIME_F.Length;

    RES = 22 - DATA_LENGTH;

    ADJ = TIME_F;

    \\* REALIZA UN AJUSTE PARA GENERAR UNA MATRIZ *///
        \\* DE 22 BITS *///
    ADJUSTMENT();

    TIME_F = ADJ;
}
```

Función *ADJUSTMENT*

```
private void ADJUSTMENT()
{
    for (int u = 0; u < RES; u++)
    {ADJ = 0 + ADJ;}
}
```

3.2.2. Interfaz para creación de TEDS

Como mínimo, un TEDS en un transductor IEEE 21451.4 deberá contener el TEDS básico que es la forma de identificar el transductor.

El TEDS básico debe estar comprendido por 64 bits, que incluyen el número de identificación del fabricante, el número del modelo, la letra de la versión, el número de la versión y el número de Serial. Estos datos deben estar organizados de acuerdo a la tabla 2.3 y contenidos en una memoria no volátil o en un TEDS virtual. También debe incluir un byte de *checksum* el cual asegura la integridad de los datos del TEDS [20].

El término “TEDS virtuales” se refiere al almacenamiento de TEDS en un sitio externo al STIM, actualmente el almacenamiento de datos puede hacerse en la nube de la Internet o en un disco duro. En los TEDS del STIM sólo se necesita almacenarse información sobre la dirección o URL del TEDS además de los TEDS básicos. El TEDS básico no tiene el requisito de la identificación del transductor y por lo tanto la URL es necesaria. La extensión de la URL debe ser el TEDS básico. (16 bytes en formato hexadecimal / ASCII) [12]. El concepto de TEDS virtuales extienden los beneficios de los TEDS estandarizados en sensores antiguos y en aplicaciones donde la memoria interna o EEPROM no está disponible.

Adicional al TEDS básico se utiliza un TEDS estándar que define las propiedades importantes para un tipo de sensor en particular. Opcionalmente, esta plantilla estándar del TEDS puede ser seguida de una plantilla de calibración. Un selector de dos bits en los datos del TEDS indican la siguiente sección. Finalmente, el final de sección de los TEDS es especificada con un área abierta para el usuario [20].

3.2. ELECCIÓN DE HARDWARE

La colección de plantillas definidas para el TEDS estándar se enumeran en la tabla 2.3. Se definen 16 plantillas para los tipos de transductores disponibles para el protocolo y 3 plantillas que corresponden a las plantillas de calibración.

En la figura 3.4 y 3.5 contiene un resumen del contenido de una plantilla estándar IEEE correspondiente a un Acelerómetro/Transductor de fuerza. Cada fila de la tabla corresponde a un comando de propiedad (designado en la plantilla con un “%”) o un comando de control que utiliza bits desde el TEDS, como una **Selección de Caso**. En la mayoría de las plantillas la segunda columna indica los diferentes casos que se pueden seleccionar en la plantilla. Por ejemplo, una selección de casos de 2 bits leído desde el TEDS puede tener 4 casos diferentes, y cada uno contiene una sección de la plantilla que es usada con su correspondiente valor del caso seleccionado.

La plantilla también indica el número de bits que se usa para cada propiedad, el nivel de acceso (usuario, calibración o identificación) y el tipo de dato. Los tipos de datos utilizados en estas plantillas estándar son resumidos en la tabla 3.1.

ConRes y ConRelRes provén un método de codificación de valores no enteros con uso mínimo de bits mediante la definición de un rango limitado sobre el cual se asignan los bits. El rango y la resolución de las propiedades ConRes y ConRelRes se enumeran en la plantilla.

3.2. ELECCIÓN DE HARDWARE

Accelerometer/Force transducer Template (ID = 25) Summary

Select	Property/Command	Description	Access	Bits	Data Type (and Range)	Units	
-	TEMPLATE	Template ID	-	8	Integer (value = 25)	-	
Select Case – Acceleration / Force				1	Select Case	-	
Select Case – Extended Functionality (Programmable Gain)				1			
Case 0 (Accel)	Case 0	%Sens@Ref	Sensitivity @ ref. condition	CAL	16	ConRelRes (5E-7 to 172, ±0.015%)	V/(m/s ²)
		%TF_HP_S	High pass cut-off freq. (F hp)	CAL	8	ConRelRes (0.005 to 13k, ±3%)	Hz
	Case 1 (Ext Funct.)	%Passive[Initialize]	Initialization not needed	ID	-	Assign = 0	-
		%Passive[CtrlFunctionMask]	Control Function Mask	ID	-	Assign = 0b11	-
		%Passive[ReadWrite]	Write only	ID	-	Assign = 3	-
		%Passive[FunctionType]	Passive control type	ID	-	Assign = 0 (checkmark)	-
		%Passive[Function]	Passive mode	USR	-	BitBin, (Assign = "xx,00")	-
		%Sens[Initialize]	Initialization not needed	ID	-	Assign = 0	-
		%Sens[CtrlFunctionMask]	Control Function Mask, Sens.	ID	-	Assign = 0b11	-
		%Sens[ReadWrite]	Write only	ID	-	Assign = 3	-
		%Sens[FunctionType]	Sensitivity control type	ID	-	Assign = 1 (one exactly)	-
		%Sens[Function]	%Sens@Ref["10"]	USR	-	BitBin, (Assign = "10")	-
		%Sens[Function]	%Sens@Ref["01"]	USR	-	BitBin, (Assign = "01")	-
		%DefaultFR	Default Sensitivity	ID	2	UNINT (00: no, 1:low, 2: high)	-
		%Passive	Supports multiplexer mode	ID	1	UNINT	-
		%Sens@Ref["01"]	Low sensitivity @ reference	CAL	16	ConRelRes (5E-7 to 172, ±0.015%)	V/(m/s ²)
		%Sens@Ref["10"]	High sensitivity @ reference	CAL	16	ConRelRes (5E-7 to 172, ±0.015%)	V/(m/s ²)
		%TF_HP_S["01"]	Low sens high pass cut-off	CAL	8	ConRelRes (0.005 to 13k, ±3%)	Hz
		%TF_HP_S["10"]	High sens high pass cut-off	CAL	8	ConRelRes (0.005 to 13k, ±3%)	Hz
		Case 1 (Force)	Select Case – Extended Functionality (Programmable Gain)				1
Case 0	%Sens@Ref		Sensitivity @ reference condition	CAL	16	ConRelRes (5E-7 to 172, ±0.015%)	V/N
	%TF_HP_S		High pass cut-off freq. (F hp)	CAL	8	ConRelRes (0.005 to 13k, ±3%)	Hz
	%Stiffness		Stiffness of transducer	CAL	6	ConRelRes (1E6 to 8.1E10, ±10%)	N/m
%Mass_below	Mass below gage		CAL	6	ConRelRes (0.1 to 8114, ±10%)	gram	
Case 1 (Ext Funct.)	%Passive[Initialize]		Initialization not needed	ID	-	Assign = 0	-
	%Passive[CtrlFunctionMask]		Control Function Mask, Passive	ID	-	Assign = 0b11	-
	%Passive[ReadWrite]		Write only	ID	-	Assign = 3	-
	%Passive[FunctionType]		Passive control type	ID	-	Assign = 0 (checkmark)	-
	%Passive[Function]		Passive mode	USR	-	BitBin, (Assign = "xx,00")	-
%Sens[Initialize]	Initialization not needed	ID	-	Assign = 0	-		

Figura 3.4: Plantilla de Acelerómetro/Transductor de fuerza (número de identificación = 25).

3.2. ELECCIÓN DE HARDWARE

		%Sens[CtrlFunctionMask]	Control Function Mask, Sens.	ID	-	Assign = 0b11	
		%Sens[ReadWrite]	Write only	ID	-	Assign = 3	-
		%Sens[FunctionType]	Sensitivity control type	ID	-	Assign = 1 (one exactly)	-
		%Sens[Function]	%Sens@Ref["10"]	USR	-	BitBin, (Assign = "10")	
		%Sens[Function]	%Sens@Ref["01"]	USR	-	BitBin, (Assign = "01")	
		%DefaultFR	Default Sensitivity	ID	2	UNINT (00: no, 1:low, 2: high)	
		%Passive	Supports multiplexer mode	ID	1	UNINT	-
		%Sens@Ref["01"]	Low sensitivity @ reference	CAL	16	ConRelRes (5E-7 to 172, ±0.015%)	V/N
		%Sens@Ref["10"]	High sensitivity @ reference	CAL	16	ConRelRes (5E-7 to 172, ±0.015%)	V/N
		%TF_HP_S["01"]	Low sens. high pass cut-off freq.	CAL	8	ConRelRes (0.005 to 13k, ±3%)	Hz
		%TF_HP_S["10"]	High sens. high pass cut-off freq.	CAL	8	ConRelRes (0.005 to 13k, ±3%)	Hz
		%Stiffness	Stiffness of transducer	CAL	6	ConRelRes (1E6 to 8.1E10, ±10%)	N/m
		%Mass_below	Mass below gage	CAL	6	ConRelRes (0.1 to 8114, ±10%)	g
		%PhaseCorrection	Phase Correction @ ref	CAL	6	ConRes (-3.2 to 3.0, step 0.1)	°
-	-	%Direction	Sensitivity direction (x, y, z)	CAL	2	Enumeration: x y z	-
-	-	%Weight	Transducer weight	CAL	6	ConRelRes (0.1 to 8114, ±10%)	gram
-	-	%ElecSigType	Transducer elect. signal type	ID	-	Assign = 0, "Voltage Sensor"	-
-	-	%MapMeth	Mapping Method	ID	-	Assign = 0, "Linear"	-
-	-	%ACDCCoupling	AC or DC coupling	ID	-	Assign = 1, "AC"	-
-	-	%Sign	Polarity (sign)	CAL	1	Enumeration: positive negative	-
Select Case – Transfer Function					1	Select Case	
Case 0	-		No transfer function specified				
Case 1	-	%TF_SP	Low pass cut-off (F lp)	CAL	7	ConRelRes (10 to 1.6E+6, ±5%)	Hz
	-	%TF_KPr	Resonance frequency (F res)	CAL	9	ConRelRes (100 to 2.4E+6, ±1%)	Hz
	-	%TF_KPq	Quality factor at F res	CAL	9	ConRelRes (0.4 to 9.7k, ±1%)	-
	-	%TF_SL	Amplitude slope (a)	CAL	7	ConRes (-6.3 to 6.3, step 0.1)	%/decade
	-	%TempCoef	Temperature Coefficient (b)	CAL	6	ConRes (-0.8 to 0.75, step 0.025)	%/°C
-	-	%Reffreq	Reference frequency	CAL	8	ConRelRes (0.35 to 2.18k, ±1.75%)	Hz
-	-	%RefTemp	Reference temperature (T ref)	CAL	5	ConRes (15 to 30, step 0.5)	°C
-	-	%CalDate	Calibration date	CAL	16	DATE	-
-	-	%CalInitials	Calibration initials	CAL	15	CHR5	-
-	-	%CalPeriod	Calibration period	CAL	12	UNINT	days
-	-	%MeasID	Measurement location ID	USR	11	UNINT	-

Figura 3.5: Plantilla de Acelerómetro/Transductor de fuerza (número de identificación = 25) continuación.

3.2. ELECCIÓN DE HARDWARE

La tabla 3.1 no es la lista completa de los tipos de datos en este estándar; solo se incluyen los tipos de datos para la plantilla estándar.

Tabla 3.1: Tipo de datos usados en las plantillas estándar de la IEEE.

Tipo de dato	Descripción
UNINT	Entero sin signo
Chr5	Carácter de 5 bits
ASCII	ASCII estándar de 7 bits
<i>Date</i>	Número de días desde enero 1, 1998
<i>Single</i>	Punto flotante de precisión simple
ConRes	Resolución constante. Este es un tipo de dato personalizado para valores de coma flotante comprimidos que proporcionan una asignación logarítmica de un intervalo definido.
ConRelRes	Resolución constante relativa. Este es un tipo de dato personalizado para valores de coma flotante comprimidos que proporcionan una asignación logarítmica de un intervalo definido.
/textitEnumeration	Hace referencia a un tipo de datos enumerado definido en la plantilla

La figura 3.6 corresponde a la interfaz que se utiliza para la creación de TEDS. En la parte izquierda se encuentra una caja de grupo con el nombre *Basic TEDS* la cual contiene los campos para recabar la información necesaria para el TEDS básico. En la parte superior de la caja de grupo se encuentra un selector de puerto para elegir entre los distintos puertos disponibles y botones para abrir, guardar o crear un nuevo documento, y en la parte inferior tenemos un selector para poder elegir y cargar el TEDS entre uno de los canales disponibles del STIM.

Un selector ubicado en la parte superior derecha permite poder elegir entre las distintas plantillas disponibles; una vez seleccionado el transductor, en la caja de

3.2. ELECCIÓN DE HARDWARE

grupo con el nombre *Standard Template* se mostrarán los campos necesarios para llenar de la plantilla correspondiente.

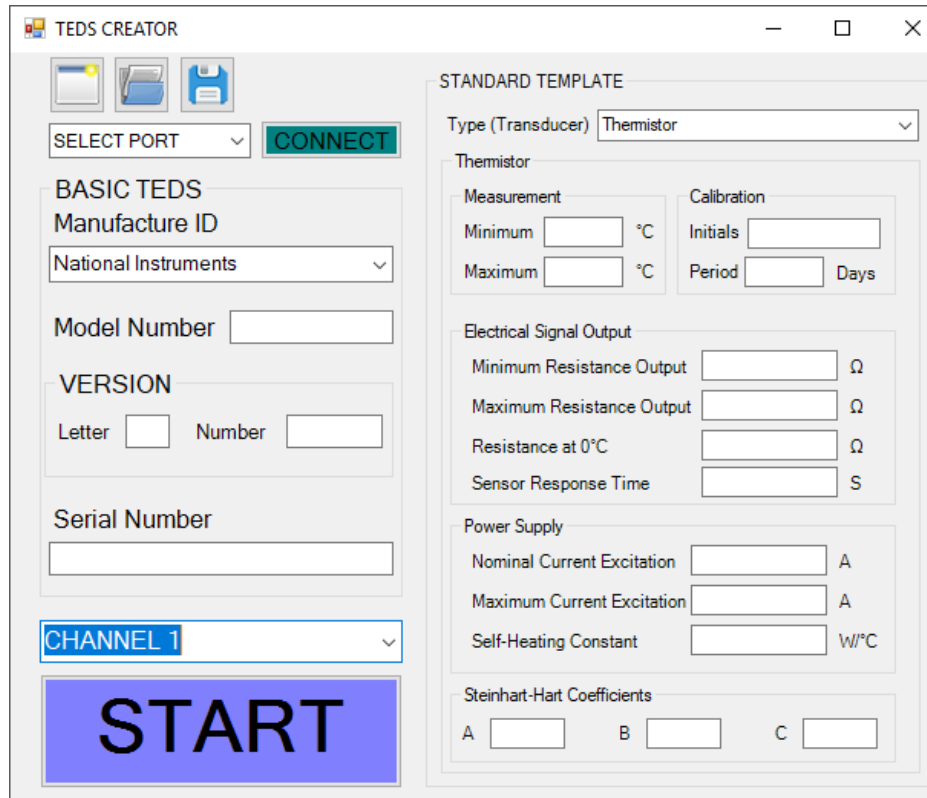


Figura 3.6: Interfaz para la creación de TEDS.

La plantilla que es mostrada es correspondiente al termistor, con número de identificación 38. La plantilla se divide en 5 partes: medición, calibración, señal eléctrica de salida, Fuente de energía y coeficientes de Steinhart-Hart, que son correspondientes a los valores para poder utilizar el sensor. Cada campo tiene su correspondiente en número entero de bits mostrados en la columna de nombre bits en la figura 3.6.

3.2. ELECCIÓN DE HARDWARE

Thermistor template (ID = 38) summary

Function	Select	Property/Command	Description	Access	Bits	Data type (and range)	Units
ID	—	TEMPLA TE	Template ID	—	8	Integer (value = 38)	—
Measurement	—	%MinPhysVal	Minimum temperature	CAL	11	ComRes (-200 to 1,846, step 1)	°C
	—	%MaxPhysVal	Maximum temperature	CAL	11	ComRes (-200 to 1,846, step 1)	°C
Electrical signal output	—	%ElecSigType	Transducer Electrical Signal Type	ID	—	Assign = 2, "Resistance Sensor"	—
	—	%MinElecVal	Minimum resistance output	CAL	18	ComRes (0 to 262k, step 1)	Ω
	—	%MaxElecVal	Maximum resistance output	CAL	18	ComRes (0 to 262k, step 1)	Ω
	—	%MapMeth	Mapping Method	ID	—	Assign = 6, "Thermistor"	—
	—	%RTDCcoef_R0	Resistance of thermistor at 0°C	ID	20	ComRelRes (10 to 5.5E+6, ±6.3ppm)	Ω
	—	%SteinhartA	Steinhart-Hart Coefficient A	ID	32	Single	1/C
	—	%SteinhartB	Steinhart-Hart Coefficient B	ID	32	Single	1/C
	—	%SteinhartC	Steinhart-Hart Coefficient C	ID	32	Single	1/C
	—	%RespTime	Sensor Response Time	ID	6	ComRelRes (1E-6 to 7.9, ±15%)	s
	Excitation or power supply	—	%ExciteAmplNom	Nominal current excitation	CAL	8	ComRelRes (1E-6 to 120E-3, ±2.3%)
—		%ExciteAmplMax	Maximum current excitation	ID	8	ComRelRes (1E-6 to 120E-3, ±2.3%)	A
—		%SelfHeating	Self-heating constant	ID	5	ComRelRes (0.25E-3 to 16E-3, ±7.4%)	W/°C
Calibration information	—	%CalDate	Calibration Date	CAL	16	DATE	—
	—	%CalInitials	Calibration initials	CAL	15	CHR5	—
	—	%CalPeriod	Calibration period	CAL	12	UNINT	days
Misc.	—	%MeasID	Measurement location ID	USR	11	UNINT	—
Total bits required for TEDS (range):							263 bits

Figura 3.7: Plantilla de termistor (número de identificación = 38).

3.2. ELECCIÓN DE HARDWARE

Al oprimir el botón *START* comenzará a obtener los valores de los campos de los grupos *Basic TEDS* y *Standard Template* para convertirlos en sus correspondientes enteros de bits. Después se envía el byte correspondiente al número de identificación de plantilla y los bits necesarios para completar la plantilla del TEDS estándar. Seguido a esto, los valores del TEDS básico, el byte de identificación y los bytes del TEDS estándar se concatenarán y serán guardados en un archivo .txt. El programa nombra al archivo con un serie de 10 números, estos se convertirán en bytes y se concatenaran para enviarlos al STIM y guardarlos en las localidades correspondientes de la EEPROM.

Como se muestra en la figura 3.7, la plantilla del TEDS estándar está compuesta por 255 bits, sin contar los bits del número de identificación de la plantilla. Al no ser un numero entero divisible entre 8 se tiene que incluir un colchón de ceros antes del bit más significativo para poder hacer que la cantidad de bits sea un numero entero de bytes. Esto se debe hacer con todos los datos que se envíen durante la comunicación del NCAP con el STIM, por ejemplo, en el caso del fin de sección donde solo se utilizan dos bits.

3.2. ELECCIÓN DE HARDWARE

3.2.2.1. Pseudocódigo

En esta sección se muestra el pseudocódigo de la función que genera los TEDS.

- Inicio de programa.
- Selección de puerto a utilizar.
- Selección el canal al cual corresponderá el TEDS.
- Presionar botón *START*.
- Iniciar función *BASIC_TEDS*.
- Iniciar función *STANDARD_TEDS*.
- Concatenar los valores del TEDS estándar según lo indica el protocolo.
- Guardar TEDS en archivo .txt y crear nombre.
- Enviar el valor representativo de la dirección de la Función.
- Enviar el valor representativo de la dirección del Canal.
- Enviar el dato del nombre del archivo .txt al STIM.
- Esperar bit de acknowledge.
- Fin del programa.

3.2. ELECCIÓN DE HARDWARE

3.2.2.2. Diagrama de flujo

La figura 3.8 muestra un diagrama de flujo que representa la secuencia de las operaciones durante la ejecución del programa para la creación del UUID.

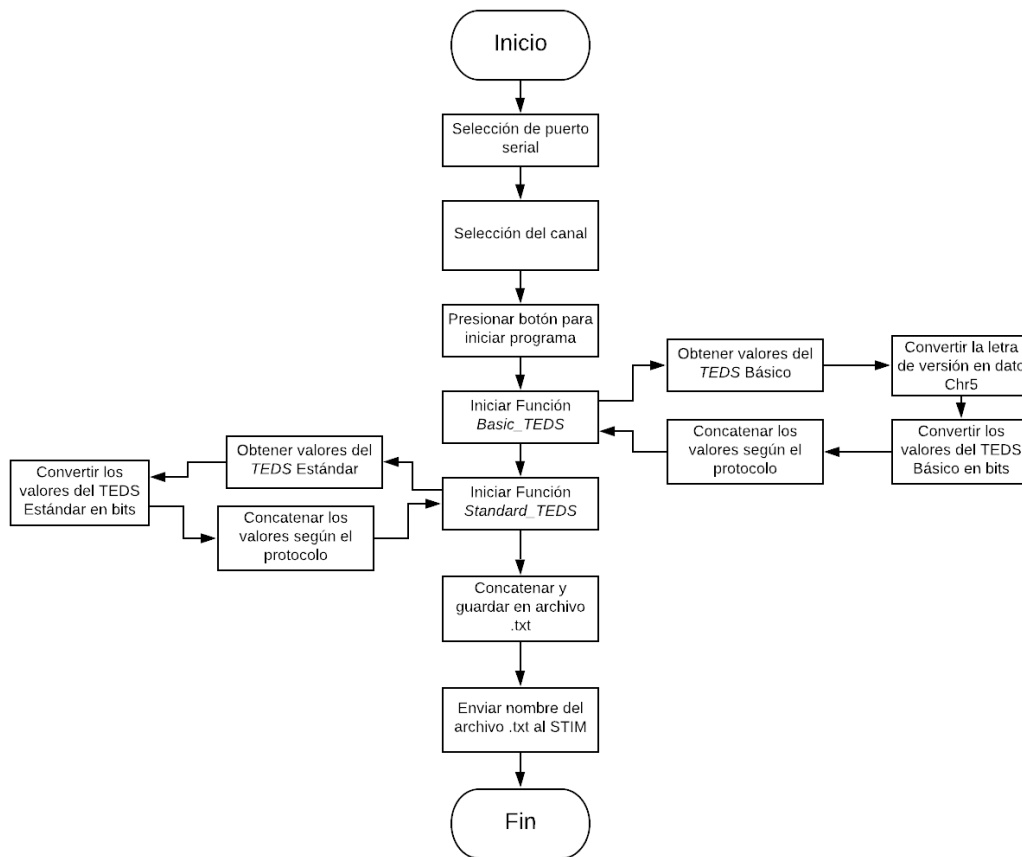


Figura 3.8: Diagrama de flujo para la creación de TEDS.

3.2. ELECCIÓN DE HARDWARE

3.2.2.3. Código de programación

Esta sección muestra parte del código que se utilizó para la interfaz creadora de TEDS.

Función *BASIC_TEDS*:

```
\\* CONVERTIR LOS DATOS DEL BASIC TEDS EN BINARIO *//
```

```
private void Basic_Teds()
```

```
{
```

```
    \\* OBTIENE EL VALOR CORRESPONDIENTE AL *//
```

```
    \\* IDENTIFICADOR DEL FABRICANTE *//
```

```
    Manuf_ID();
```

```
    \\* OBTIENE LOS VALORES DE LOS CAMPOS *//
```

```
    \\* Y LOS CONVIERTE EN ENTEROS *//
```

```
    int Model_Number = Mod_num.Text.Length > 0 ?
```

```
        int.Parse(Mod_num.Text) : 0;
```

```
    int Version_Number=Ver_Num.Text.Length > 0 ?
```

```
        int.Parse(Ver_Num.Text) : 0;
```

```
    int Serial_Number=Serial_num.Text.Length > 0 ?
```

```
        Convert.ToUInt16(Serial_num.Text, 16) : 0;
```

3.2. ELECCIÓN DE HARDWARE

```
\\*OBTIENE EL VALOR DEL CAMPO VERSION LETTER*//  
\\* Y LOS CONVIERTE EN DATO CHR5*//  
V_Let = Ver_letter.Text;  
  
Version_LETTER();  
  
BasicTeds = "";  
  
TEDS = "";  
  
int a = 0;  
  
\\*CONVIERTE Y CONCATENA LOS VALORES EN*//  
\\*UN CADENA DE 64 BITS*//  
while (a <64)  
{  
    while (a < 14)  
    {  
        BasicTeds = Manufacture_ID % 2 +  
        BasicTeds;  
  
        Manufacture_ID = Manufacture_ID / 2;
```

3.2. ELECCIÓN DE HARDWARE

```
        a++;
    }
    TEDS = TEDS + BasicTeds;

    BasicTeds = "";

    while ((a >= 14) && (a < 29))
    {
        BasicTeds = Model_Number % 2 +
        BasicTeds;

        Model_Number = Model_Number / 2;

        a++;
    }
    TEDS = TEDS + BasicTeds;

    BasicTeds = "";

    while ((a >= 29) && (a < 34))
    {
        BasicTeds = Version_Letter % 2 +
        BasicTeds;
```

3.2. ELECCIÓN DE HARDWARE

```
Version_Letter = Version_Letter / 2;

    a++;
}
TEDS = TEDS + BasicTeds;

BasicTeds = "";

while ((a >= 34) && (a < 40))
{
    BasicTeds = Version_Number % 2 +
    BasicTeds;

    Version_Number = Version_Number / 2;

    a++;
}
TEDS = TEDS + BasicTeds;

BasicTeds = "";

while ((a >= 40) && (a < 64))
```

3.2. ELECCIÓN DE HARDWARE

```
    {  
        BasicTeds = Serial_Number % 2 +  
        BasicTeds ;  
  
        Serial_Number = Serial_Number / 2 ;  
  
        a++ ;  
    }  
    TEDS = TEDS + BasicTeds ;  
  
    Save_TEDS = Save_TEDS + TEDS ;  
  
    LENGTH = Save_TEDS.Length ;  
}
```

Función *Manuf_ID*:

```
\\*OBTENER EL DATO DEL IDENTIFICADOR DE FABRICANTE*//  
public void Manuf_ID()  
    { \\* VER TABLAS A.2 , A.3 Y A.4*//  
    \\*SWITCH CASE CON LOS VALORES DE ID DE FABRICANTE*//  
        switch (Manfac_Id){  
            case 0:    Manufacture_ID = 17; break ;  
            case 1:    Manufacture_ID = 18; break ;
```


3.2. ELECCIÓN DE HARDWARE

```
        case 2:  Manufacture_ID = 19; break;
        case 3:  Manufacture_ID = 20; break;
        case 4:  Manufacture_ID = 21; break;
        .
        .
        .
        case 127: Manufacture_ID = 15095; break;
        case 128: Manufacture_ID = 15913; break;
        case 129: Manufacture_ID = 16284; break;
        default: Manufacture_ID = 0; break;
    }
}
```

Función *STANDARD_TEDS*:

```
\\* CONVERTIR LOS DATOS DEL STANDARD TEDS EN BINARIO *//
private void Version_LETTER()
{
    \\* VER TABLAS A.1 *//
    \\* SWITCH CASE CON LOS VALORES DE VERSION LETTER *//
    switch (V_Let)
    {
        case " ": Version_Letter = 0; break;
        case "a": Version_Letter = 1; break;
        case "b": Version_Letter = 2; break;
    }
}
```

3.2. ELECCIÓN DE HARDWARE

```
        case "c": Version_Letter = 3; break;
        .
        .
        .
        case "/": Version_Letter = 29; break;
        case "_": Version_Letter = 30; break;
        case "@": Version_Letter = 31; break;
        default: MessageBox.Show("ERROR"); break;
    }
}
```

Función *STANDARD_TEDS*:

```
\\* CONVERTIR LOS DATOS DEL STANDARD TEDS EN BINARIO *//
private void Standard_Teds()
{ \\* INICIAR VARIABLES *//
    StandardTeds = "";
    int m = 0;
    int MinPhysVal;
    int MaxPhysVal;
    int MinElecVal;
    int MaxElecVal;
    int ExciteAmplNom;
    int ExciteAmplMax;
```

3.2. ELECCIÓN DE HARDWARE

```
int CalDate;
int CalInitials;
int CalPeriod;
int MeasID;
int template_id=Template;
StdTeds = "";
StandardTeds = "";
Type_id = "";
m = 0;

\\* CONVERTIR A BITS EL VALOR DEL *//
\\* IDENTIFICADOR DE PLANTILLA *//
while (m < 8)
{
    Type_id = template_id % 2 + Type_id;

    template_id = template_id / 2;

    m++;
}

LENGTH = Type_id.Length;
TEDS = TEDS+Type_id;
```

3.2. ELECCIÓN DE HARDWARE

```
\\*SWITCH CASE CON LA PLANTILLA DE TRANSDUCTOR*//  
    \\* DEL TERMISTOR *//  
  
case 38:    // THERMISTOR //  
  
    \\*OBTIENE LOS VALORES DE LOS CAMPOS*//  
    \\* Y LOS CONVIERTE EN ENTEROS *//  
    MinPhysVal = Min_Phy_Thermistor.Text.Length > 0 ?  
    int.Parse(Min_Phy_Thermistor.Text) : 0;  
  
    MaxPhysVal = Max_Phy_Thermistor.Text.Length > 0 ?  
    int.Parse(Max_Phy_Thermistor.Text) : 0;  
  
    MinElecVal = Min_Elec_Thermistor.Text.Length > 0 ?  
    int.Parse(Min_Elec_Thermistor.Text) : 0;  
  
    MaxElecVal = Max_Elec_Thermistor.Text.Length > 0 ?  
    int.Parse(Max_Elec_Thermistor.Text) : 0;  
  
    int RTDCoef_R0 = RTD_R0_Thermistor.Text.Length > 0 ?  
    int.Parse(RTD_R0_Thermistor.Text) : 0;
```

3.2. ELECCIÓN DE HARDWARE

```
int SteinhartA = Sth_A_Thermistor.Text.Length > 0 ?  
int.Parse(Sth_A_Thermistor.Text) : 0;
```

```
int SteinhartB = Sth_B_Thermistor.Text.Length > 0 ?  
int.Parse(Sth_B_Thermistor.Text) : 0;
```

```
int SteinhartC = Sth_C_Thermistor.Text.Length > 0 ?  
int.Parse(Sth_C_Thermistor.Text) : 0;
```

```
RespTime = Resp_Time_Thermistor.Text.Length > 0 ?  
int.Parse(Resp_Time_Thermistor.Text) : 0;
```

```
ExciteAmplNom = Exc_Nom_Thermistor.Text.Length > 0 ?  
int.Parse(Exc_Nom_Thermistor.Text) : 0;
```

```
ExciteAmplMax = Exc_Max_Thermistor.Text.Length > 0 ?  
int.Parse(Exc_Max_Thermistor.Text) : 0;
```

```
int SelfHeating = Self_Heat_Thermistor.Text.Length > 0 ?  
int.Parse(Self_Heat_Thermistor.Text) : 0;
```

```
CalDate = 1;
```

3.2. ELECCIÓN DE HARDWARE

```
CalInitials = Cal_Initials_Thermistor.Text.Length > 0 ?
int.Parse(Cal_Initials_Thermistor.Text) : 0;
```

```
CalPeriod = Cal_Period_Thermistor.Text.Length > 0 ?
int.Parse(Cal_Period_Thermistor.Text) : 0;
```

```
MeasID = 1;
```

```
\\*CONVIERTE Y CONCATENA LOS VALORES EN*///
```

```
\\*UN CADENA DE BITS *///
```

```
m = 0;
while (m < 11)
{
    StandardTeds = MinPhysVal % 2 + StandardTeds;
    MinPhysVal = MinPhysVal / 2;
    m++;
}
StdTeds = StdTeds + StandardTeds;
```

```
StandardTeds = "";
```

```
m = 0;
while (m < 11)
{
```

3.2. ELECCIÓN DE HARDWARE

```
StandardTeds = MaxPhysVal % 2 + StandardTeds ;
MaxPhysVal = MaxPhysVal / 2 ;
m++;
}
StdTeds = StdTeds + StandardTeds ;

StandardTeds = "";
m = 0;
while (m < 18)
{
StandardTeds = MinElecVal % 2 + StandardTeds ;
MinElecVal = MinElecVal / 2 ;
m++;
}
StdTeds = StdTeds + StandardTeds ;

StandardTeds = "";
m = 0;
while (m < 18)
{
StandardTeds = MaxElecVal % 2 + StandardTeds ;
MaxElecVal = MaxElecVal / 2 ;
m++;
```

3.2. ELECCIÓN DE HARDWARE

```
    }
    StdTeds = StdTeds + StandardTeds;

    StandardTeds = "";
    m = 0;
    while (m < 20)
    {
        StandardTeds = RTDCoef_R0 % 2 + StandardTeds;
        RTDCoef_R0 = RTDCoef_R0 / 2;
        m++;
    }
    StdTeds = StdTeds + StandardTeds;

    StandardTeds = "";
    m = 0;
    while (m < 32)
    {
        StandardTeds = SteinhartA % 2 + StandardTeds;
        SteinhartA = SteinhartA / 2;
        m++;
    }
    StdTeds = StdTeds + StandardTeds;
```


3.2. ELECCIÓN DE HARDWARE

```
StandardTeds = "";
m = 0;
while (m < 32)
{
    StandardTeds = SteinhartB % 2 + StandardTeds;
    SteinhartB = SteinhartB / 2;
    m++;
}
StdTeds = StdTeds + StandardTeds;
```

```
StandardTeds = "";
m = 0;
while (m < 32)
{
    StandardTeds = SteinhartC % 2 + StandardTeds;
    SteinhartC = SteinhartC / 2;
    m++;
}
StdTeds = StdTeds + StandardTeds;
```

```
StandardTeds = "";
m = 0;
while (m < 6)
```

3.2. ELECCIÓN DE HARDWARE

```
{
    StandardTeds = RespTime % 2 + StandardTeds;
    RespTime = RespTime / 2;
    m++;
}
StdTeds = StdTeds + StandardTeds;

StandardTeds = "";
m = 0;
while (m < 8)
{
    StandardTeds = ExciteAmplNom % 2 + StandardTeds;
    ExciteAmplNom = ExciteAmplNom / 2;
    m++;
}
StdTeds = StdTeds + StandardTeds;

StandardTeds = "";
m = 0;
while (m < 8)
{
    StandardTeds = ExciteAmplMax % 2 + StandardTeds;
    ExciteAmplMax = ExciteAmplMax / 2;
```

3.2. ELECCIÓN DE HARDWARE

```
        m++;
    }
    StdTeds = StdTeds + StandardTeds;

    StandardTeds = "";
    m = 0;
    while (m < 5)
    {
        StandardTeds = SelfHeating % 2 + StandardTeds;
        SelfHeating = SelfHeating / 2;
        m++;
    }
    StdTeds = StdTeds + StandardTeds;

    StandardTeds = "";
    m = 0;
    while (m < 16)
    {
        StandardTeds = CalDate % 2 + StandardTeds;
        CalDate = CalDate / 2;
        m++;
    }
    StdTeds = StdTeds + StandardTeds;
```

3.2. ELECCIÓN DE HARDWARE

```
StandardTeds = "";
m = 0;
while (m < 15)
{
    StandardTeds = CalInitials % 2 + StandardTeds;
    CalInitials = CalInitials / 2;
    m++;
}
StdTeds = StdTeds + StandardTeds;
```

```
StandardTeds = "";
m = 0;
while (m < 12)
{
    StandardTeds = CalPeriod % 2 + StandardTeds;
    CalPeriod = CalPeriod / 2;
    m++;
}
StdTeds = StdTeds + StandardTeds;
```

```
StandardTeds = "";
m = 0;
```

3.2. ELECCIÓN DE HARDWARE

```
while (m < 11)
{
    StandardTeds = MeasID % 2 + StandardTeds;
    MeasID = MeasID / 2;
    m++;
}
StdTeds = StdTeds + StandardTeds;

LENGTH = TEDS.Length;

break;

default:

    MessageBox.Show("ERROR");

    break;
}

}
```

3.2. ELECCIÓN DE HARDWARE

3.2.3. Interfaz para la visualización de datos, NCAP

El NCAP proporciona un control para la pasarela de información que existe entre los diferentes STIM conectados.

La interfaz del NCAP se desarrolla un programa libre. Contiene diferentes campos para mostrar la información del STIM y del transductor seleccionado. La figura 3.9 contiene esta interfaz; la parte izquierda contiene la información del STIM donde estamos conectado, en la parte inferior izquierda podemos seleccionar entre los distintos canales disponibles del STIM.

Una vez conectado el STIM y seleccionado el canal podemos observar en el lado derecho una gráfica en tiempo real que nos muestra los valores medidos. En la parte inferior muestra los datos del transductor comenzando por el TEDS básico y luego el TEDS estándar.

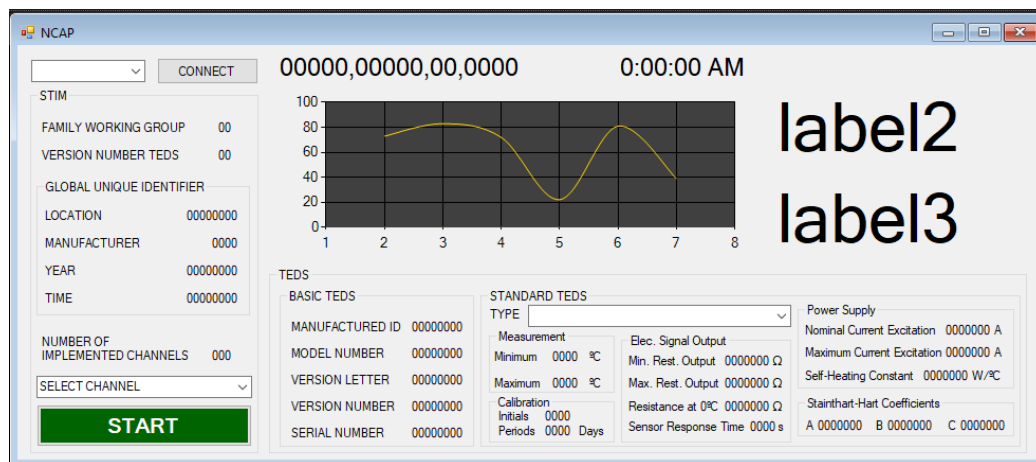


Figura 3.9: Interfaz del NCAP.

3.2. ELECCIÓN DE HARDWARE

3.2.3.1. Pseudocódigo

- Inicio de programa.
- Selección de puerto a utilizar.
- Enviar Dirección de Función.
- Enviar Dirección del Canal 0.
- Leer datos del STIM (UUID).
- Mostrar datos del STIM (UUID).
- Leer datos de los canales disponibles del STIM.
- Selección del canal disponible.
- Enviar Dirección de Función.
- Enviar Dirección del Canal Seleccionado.
- Lectura de datos del nombre del archivo contenedor del TEDS.
- Lectura del TEDS básico.
- Lectura del TEDS estándar.
- Imprimir datos en pantalla.
- Enviar función para comenzar lectura de los datos de la medición.
- Enviar función para detener lectura.
- Fin del programa.

3.2. ELECCIÓN DE HARDWARE

3.2.3.2. Diagrama de flujo

La figura 3.10 muestra un diagrama de flujo que representa la secuencia de las operaciones durante la ejecución del programa para el NCAP.

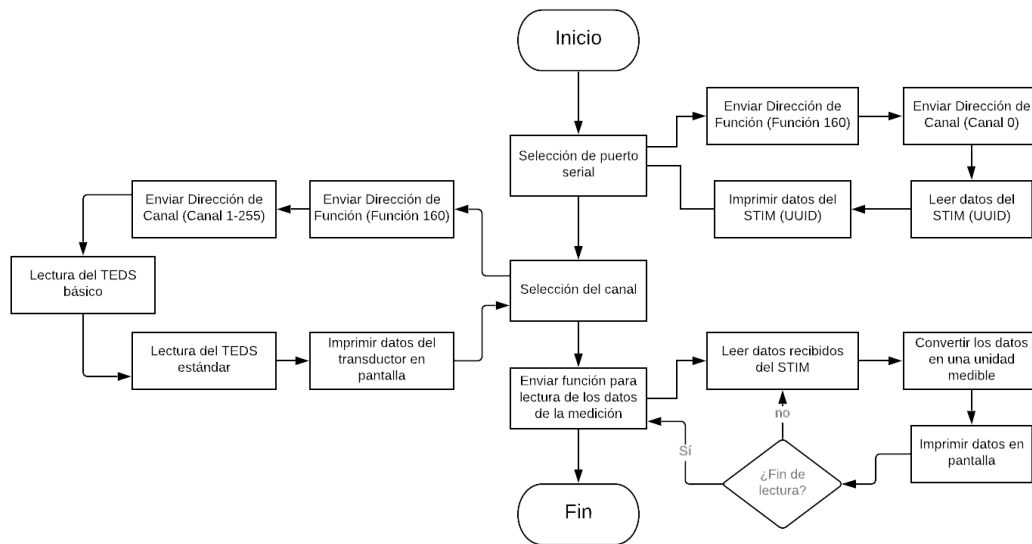


Figura 3.10: Diagrama de flujo para el NCAP.

Capítulo 4

Pruebas experimentales y resultados del Protocolo IEEE 21451

En este capítulo se describe el desarrollo experimental del sistema basado en una computadora que funcionará como NCAP y un arduino Mega 2560, utilizado como módulo de adquisición de datos analógicos para simular un STIM que es capaz de soportar hasta 16 transductores. Se diseñó un sistema de TEDS Clase II debido a que el arduino establece dos líneas de comunicación, una línea analógica, que corresponde a los pines análogos donde se conectan los transductores y el puerto serial, una línea digital conectada al NCAP.

El arduino tiene varios pines de entrada analógicas y digitales para la adquisición de datos, además de eso dispone de un puerto USB tipo B con el que fácilmente se conecta a la computadora para iniciar la comunicación serial. El protocolo establece distintas formas para la comunicación digital del STIM, en este proyecto se utiliza la comunicación serial en el STIM, al igual que en las interfaces desarrolladas, esto para hacer más sencillo el envío de los datos, sin embargo, se pueden acoplar las interfaces a otra forma de comunicación, como por ejemplo

4.1. ELECCIÓN DE HARDWARE

bluetooth y lograr el envío de los datos sin ningún problema.

El proyecto se basa en 3 interfaces creadas en Visual Basic. La interfaz creadora de UUID y la interfaz creadora de TEDS envían datos a través de la conexión digital hacia el arduino y el arduino se encarga de almacenar esta información en su memoria EEPROM interna. La tercera interfaz es el NCAP, esta es capaz de enviar señales al STIM y a sus canales para la activación y control de los mismos, además de leer la información que es enviada por el arduino y mostrarlo en pantalla para el usuario.

4.1. Elección de Hardware

Se utilizó una laptop HP Newest jaguar 15 con procesador AMD A10-9600P RADEON R5 @ 2.40GHz con memoria RAM de 16.0 GB, con un sistema operativo *Windows* 10 de 64-bits.

Se utilizó una placa *arduino Mega 2560*, siendo probablemente el microcontrolador más capaz de la familia, el cual dentro de sus características posee 54 pines digitales que funcionan como entrada/salida [1]; 16 entradas análogas en las cuales podemos hacer la adquisición de datos analógicos que funcionarán como nuestros canales del STIM, una EEPROM de 4KB para el almacenamiento de los dato para los TEDS, y, además, una conexión USB.

4.2. STIM

La estructura del STIM no viene definida dentro del protocolo, esto quiere decir que el fabricante del STIM puede desarrollarlo de la manera que crea que es más conveniente. El STIM debe de constar de un microcontrolador, ingeniería de conversión de señal puertos de entrada y de salida por lo que se puede desarrollar en una placa de hardware y software libre como lo es el ARDUINO MEGA 2560.

El programa que se desarrolló para el STIM consta de varios *SWITCH CASE* los cuales se seleccionarán según la dirección de función que envié el NCAP.

4.2.1. Escritura en la EEPROM del Arduino

Lo que se pretende hacer es simular un módulo STIM con la placa arduino, que sea capaz de realizar una operación de escritura en la memoria EEPROM a través del puerto serial. La memoria EEPROM son de alta capacidad, además de que no se tiene que desconectar de la placa para cargar los archivos, lo que facilita realmente su uso.

El proyecto muestra dos interfaces, la interfaz creadora de UUID y la creadora de los TEDS. La primera convierte los valores de longitud, latitud, la fecha de creación y de manufactura del STIM en diez bytes que se envían a través del puerto serial, escribiendo la información generada por el UUID en la memoria EEPROM del arduino. La figura 4.1 muestra un ejemplo del uso de la interfaz para crear TEDS. La conversión de los datos se muestra en el apartado 3.2.1.

El creador de TEDS da un número indefinido de bytes que se almacenan en un

4.2. STIM

archivos .txt; esto se debe a que a pesar de que el TEDS básico es un constante de 8 bytes, el TEDS estándar varía dependiendo del transductor utilizado. El creador de TEDS también es el encargado de escribir los bytes correspondientes al nombre del archivo generado en la EEPROM del STIM. La conversión de los datos de los TEDS se muestra en el apartado 3.2.2.

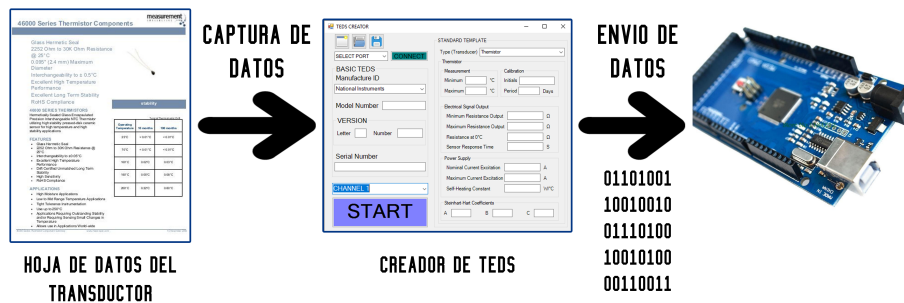


Figura 4.1: Uso de la interfaz para la creación de TEDS.

Al conectar el arduino a través del puerto serial e iniciado el programa para la creación de UUID o TEDS, se tienen que capturar los datos de los campos de la interfaz y seleccionar el puerto en el que está conectado el STIM para que se inicie la conversión y el envío de los datos. La interfaz se encarga de recopilar todos los datos de los campos mostrados y convertirlos en bytes que se puedan enviar a través del puerto serial.

El arduino con su función *Serial.read* [3] se encarga de leer los datos recibidos desde la computadora y separa los datos en bytes. Los primeros dos bytes indican la función y el canal que se utilizan durante la escritura de los datos.

En el caso de enviar datos desde el creador de UUID, el STIM tendrá que estar preparado para recibir solo 10 bytes. Si recibe datos desde el creador de

4.2. STIM

TEDS, el STIM tendrá que recibir 1 byte correspondiente al canal seleccionado y luego esperar los siguientes 10 bytes que corresponden al nombre del archivo .txt generado por la interfaz. Una vez recibidos los datos del puerto serial la función *EEPROM.update* [2] tomará los bytes generados por las interfaces y los almacena en las localidades disponible de la EEPROM. Utilizando la librería de arduino para la EEPROM [2] fácilmente se tiene acceso a la memoria interna para guardar los datos de nuestros TEDS.

Para poder corroborar que la información que se guardó en la EEPROM es la correcta se propuso un programa que muestras las localidades de la EEPROM y los datos en el monitor serial. La figura 4.2 muestra el programa y los valores contenidos en la EEPROM.

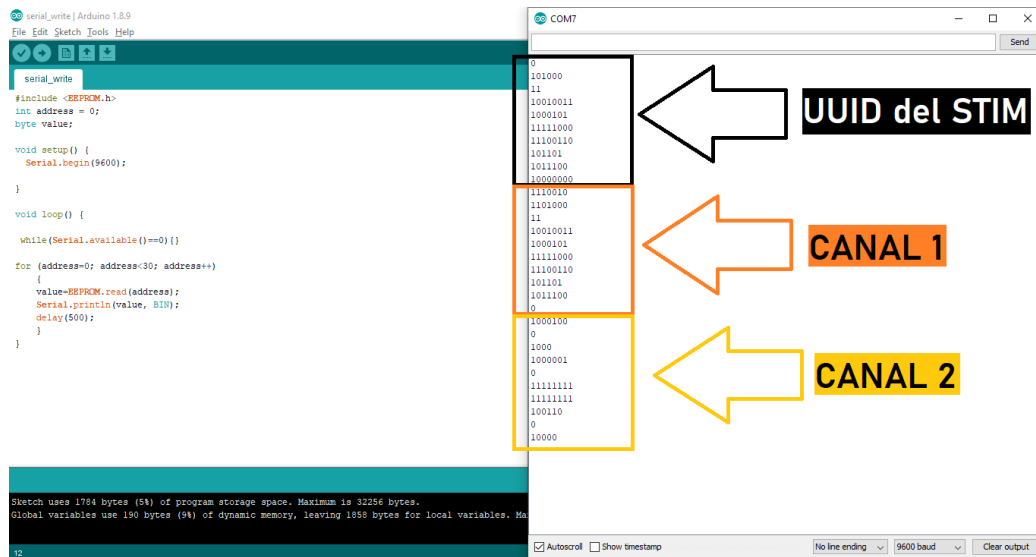


Figura 4.2: Datos contenidos en la EEPROM.

Los primeros 10 bytes corresponden a la información del UUID del STIM,

4.2. STIM

seguidos por los 10 bytes correspondientes al nombre del archivo del sensor en el primer canal y consecutivamente los 10 bytes del nombre del archivo del sensor en el segundo canal. La información de los distintos transductores se guarda en los siguientes campos de la EEPROM, siempre con un tamaño de 10 bytes que corresponderán a un nombre de un archivo guardado en la PC.

Como resumen, para realizar la escritura del nombre del archivo contenedor de TEDS en la memoria hay que seguir estos pasos:

- Instalar en el STIM el programa adecuado para la comunicación y la escritura de los archivos para los TEDS.
- Una vez realizada la carga del programa, con un software apropiado creado en Visual Basic, podremos iniciar la escritura de los datos del TEDS. Para este trabajo se ha utilizado la interfaz que se muestra en el apartado 3.4.2 donde se puede introducir todos los datos del TEDS básico y del TEDS estándar.
- Mediante esta interfaz se puede grabar el nombre del archivo generado, en donde está la información del transductor seleccionado, a través de un puerto serie.

4.2.2. Lectura desde el Arduino

Una vez hecha la escritura se procederá a la validación de ésta. Para realizar la lectura se emplea la misma conexión que utilizamos durante la escritura, sin embargo, para este caso debemos tener instalado la interfaz del NCAP. La

4.2. STIM

interfaz creada en Visual Basic para recepción de TEDS muestra en pantalla la información que corresponde al canal seleccionado del STIM.

Una vez que las interfaces han cargado los datos al arduino, el NCAP envía la función al STIM para leer esta información y comenzar la lectura del transductor que se encuentra en el canal seleccionado.

La comunicación entre NCAP y STIM se establece por el puerto serial del arduino. El STIM utiliza la función *EEPROM.read*, para leer los datos guardados en las localidades de la EEPROM, y con *Serial.write*, envía los datos como bytes a través del puerto serial. El NCAP recibe estos datos comenzando con los datos del UUID, seguidos por los datos que corresponden al nombre del archivo que contiene el TEDS del canal seleccionado.

Los bytes recibidos se almacenan en matrices para obtener los valores reales y mostrarlos en los campos disponibles del NCAP. Los TEDS virtuales contendrán los datos del TEDS básico, el número de identificación y el TEDS estándar. El NCAP tendrá acceso a ella a través de su descarga desde la PC utilizando el nombre del archivo correspondiente al canal que se está utilizando. Una vez recibidos los datos el NCAP le envía la señal al STIM que comience la recopilación o el envío de datos del transductor que se encuentre en el canal seleccionado.

4.2.2.1. Pseudocódigo

En este apartado se muestran los pseudocódigos para la escritura y la lectura de los TEDS.

4.2. STIM

Escritura del TEDS

Los datos del TEDS se guardaran en un archivo con extensión .txt. Para reconocer entre los diferentes archivos de cada canal la interfaz creara un nombre de 10 bytes que se almacenaran en las localidades del STIM.

- Inicio de programa.
- Obtener valores de los campos del TEDS básico y del TEDS estándar.
 - TEDS básico (64 bits)
 - Número de identificación de la plantilla (8 bits).
 - TEDS estándar (cantidad de bits marcado en la plantilla mas colchón de ceros).
- Crear archivo .txt
- Guardar los datos de los TEDS.
- Crear nombre de archivo.
- Leer Dirección de Función (Función 0).
- Leer Dirección de Canal (Canal 1-255).
- Recibir matriz de datos
- Separar matriz de datos
- Guardar los datos en las localidades disponibles de la EEPROM.
- Fin del programa.

4.2. STIM

Lectura del TEDS

Para obtener los datos de los TEDS la interfaz del NCAP hace lectura de los bytes almacenados en la EEPROM del STIM y abre el archivo .txt correspondiente.

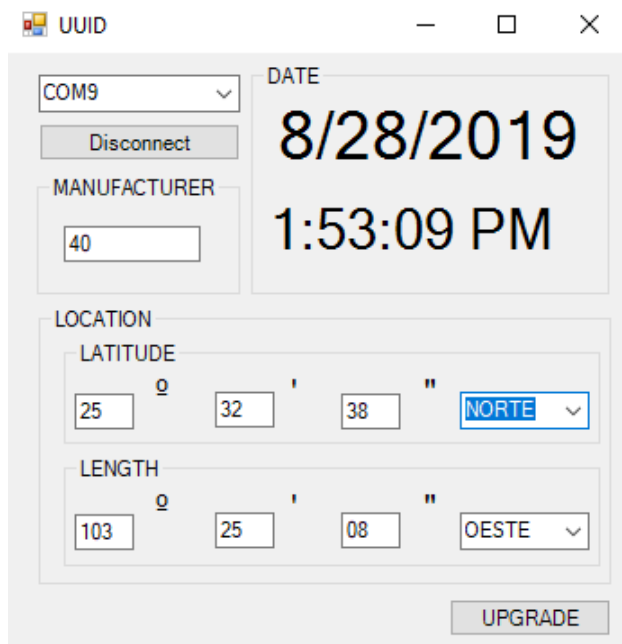
- Inicio de programa.
- Leer Dirección de Función (Función 160).
- Leer Dirección de Canal (Canal 1-255).
- Enviar datos de las localidades correspondientes al nombre del archivo contenedor del TEDS del canal.
- Abrir el archivo contenedor del TEDS.
 - TEDS básico (64 bits, leídos de la EEPROM).
 - Numero de identificación de la plantilla (8 bits, leídos de la EEPROM).
 - TEDS estándar (cantidad de bits marcado en la plantilla, leídos de la EEPROM)
- Fin del programa.

4.2.3. Pruebas del módulo STIM y del NCAP

Las pruebas se realizaron simulando un termistor NTC de encapsulado de vidrio de la serie 44306 del fabricante *Measurement Specialties*. Para poder hacer

4.2. STIM

las pruebas de la comunicación entre el STIM y la NCAP primero se debe cargar la información de UUID y de los TEDS al STIM. La figura 4.3 muestra los datos capturados del UUID que son enviados al STIM.



The screenshot shows a software window titled "UUID" with standard Windows window controls (minimize, maximize, close). The interface is divided into several sections:

- COM9**: A dropdown menu showing "COM9".
- Disconnect**: A button.
- MANUFACTURER**: A text input field containing "40".
- DATE**: A large digital display showing "8/28/2019" and "1:53:09 PM".
- LOCATION**: A section containing:
 - LATITUDE**: Three input fields with values "25", "32", and "38", followed by a dropdown menu set to "NORTE".
 - LENGTH**: Three input fields with values "103", "25", and "08", followed by a dropdown menu set to "OESTE".
- UPGRADE**: A button at the bottom right.

Figura 4.3: Captura de datos para el UUID.

Una vez cargado los datos del UUID se capturan los datos del termistor y se envían a través del puerto serial. La figura 4.4 muestra los datos capturados del termistor. Terminado el envío de los datos se comprueban que estos hayan sido correctos con el programa descrito en el apartado 4.1.1.

Al iniciar la comunicación entre NCAP y STIM, el NCAP envía la señal para recibir la información de la UUID del STIM contenida en las primeras 10 localidades de la EEPROM. Además, recibirá el bloque de los datos del Meta-TEDS del STIM descritos en las tablas 2.10 y 2.11. Una vez seleccionado el canal donde

4.2. STIM

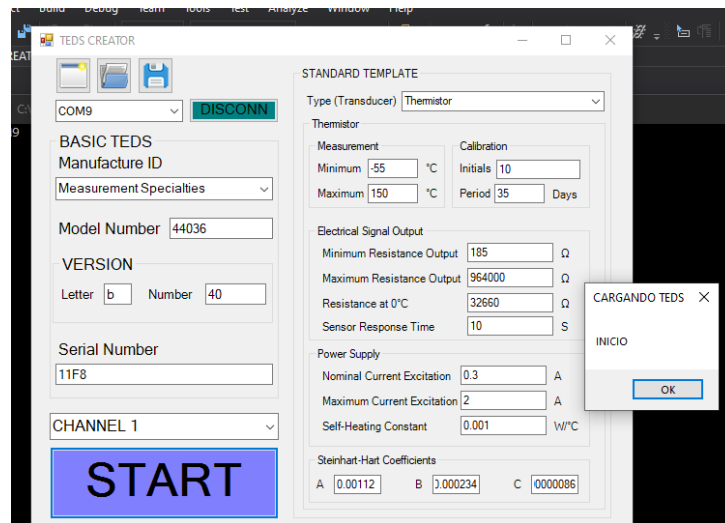


Figura 4.4: Captura de datos para el creador de TEDS.

está conectado el transductor, el NCAP recibiría el bloque de datos del TEDS de este canal descritos en la tabla 2.12 y 2.13. La simulación del sensor se realizó conectando un potenciómetro al pin analógico 0 del STIM como se muestra en la figura 4.5 para observar mayor variación en la temperatura.

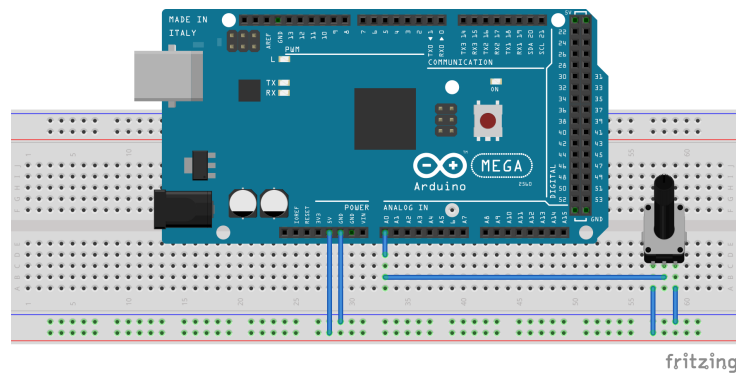


Figura 4.5: Diagrama de conexión de potenciómetro.

Las pruebas con el módulo muestran que la comunicación con el NCAP y

4.2. STIM

el envío de los datos del UUID y de los TEDS se realizan de manera correcta. En la figura 4.6 se observa los datos recibidos desde el STIM. La parte izquierda corresponde a la información del STIM y en la parte inferior derecha se tienen los datos del TEDS básico y del TEDS estándar. Las variaciones de la información en el canal seleccionado se detectan de manera oportuna, y la gráfica en tiempo real mostrada en la parte superior derecha del NCAP corresponde correctamente a dichas variaciones.

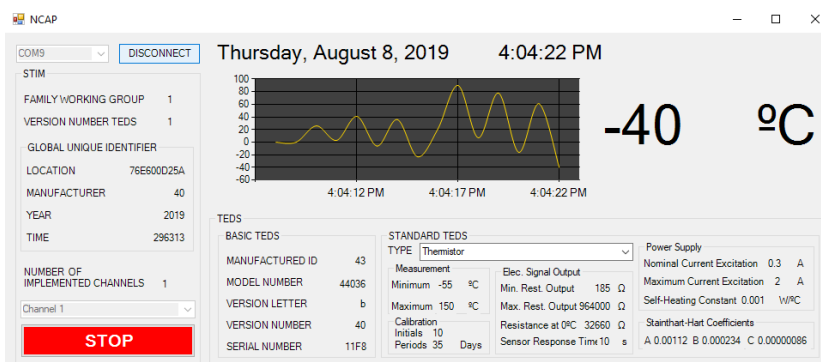


Figura 4.6: Demostración del funcionamiento del NCAP.

Se analizaron estos resultados para determinar que la implementación del protocolo es la adecuada, además de que se pueden realizar pruebas con otros sensores y permite conocer la facilidad con la que el protocolo puede auxiliar al usuario para compartir diferente información de distintos sensores. Se pretende que esta investigación puede generar avances para ser aplicada en proyectos futuros.

Capítulo 5

Conclusión y Recomendaciones

5.1. Conclusión

En este escrito se propuso un diseño de un módulo desarrollado en una tarjeta arduino e interfaces para el control y la comunicación de los transductores inteligentes para validar el protocolo 21452 de la IEEE.

Se desarrolló una interfaz para la creación y envío de los datos del TEDS. Asimismo, se desarrolló una interfaz para la creación y envío de los datos del Identificador Único Universal, UUID, hacia el STIM. Al realizar las pruebas se concluye que los resultados son los esperados y que la escritura de los TEDS y del UUID en la memoria EEPROM del arduino es satisfactoria. También se demuestra que la información de los TEDS puede compartirse de manera más efectiva con ayuda de este protocolo.

Además, se muestra una interfaz para el NCAP, que permite al usuario la comunicación y control entre los distintos STIM disponibles. Haciendo uso de esta interfaz podemos observar que los datos recibidos en el NCAP se muestran de

5.1. CONCLUSIÓN

manera correcta y los datos obtenidos por el módulo son correctos.

Se propone un pseudocódigo para el desarrollo del STIM en una placa electrónica de hardware libre que incorpore un microcontrolador reprogramable. El proyecto opta por un sistema de TEDS de Clase II debido a las características de los instrumentos utilizados en este proyecto, siendo funcional y entregando los datos de manera correcta. Sin embargo, en un futuro se podría desarrollar en un sistema de TEDS Clase I el cual, a pesar de ser más complejo, tiene ventajas al no necesitar dos canales para la adquisición de datos (adquisición de datos digitales y analógicos). La implementación del protocolo utilizando una PC, un microcontrolador y un sensor analógico, demuestran que los sistemas de *plug&play* del protocolo 21451 propuesto en la IEEE para transductores inteligentes que se menciona al principio son eficientes y obtienen grandes ventajas de uso comparados con la forma usual de programar los transductores. Asimismo, este protocolo puede desarrollarse fácilmente en diferentes plataformas, sistemas, placas de computadoras o microcontroladores siguiendo los pasos de los pseudocódigos propuestos en el capítulo 3, lo que demuestra que la reconversión de los actuales sensores analógicos a sensores inteligentes implicaría un beneficio mucho mayor al coste que implica la conversión no solamente en el aspecto económico, ya que siempre que hay un problema de error en la calibración el costo de reparación es alto por no hablar de las posibles consecuencias que los fallos pudieran generar; sino, los beneficios en cuanto a la seguridad que generarían estos sensores.

Evidentemente el mayor beneficio es para los fabricantes de transductores al desarrollar módulos de transductores inteligentes que utilicen TEDS, ya que per-

5.2. RECOMENDACIONES

mite centrarse únicamente en el diseño de los transductores dejando a un lado compatibilidad con las distintas formas de comunicación.

5.2. Recomendaciones

El Arduino Mega 2560 cuenta con recursos suficientes para poder controlar 16 transductores, lo que lo hace ideal para realizar pruebas de laboratorio del protocolo, pero se recomienda utilizar un STIM que tenga una mayor cantidad de entradas analógicas para que el usuario pueda extenderse hasta la máxima capacidad de transductores que permite el protocolo controlar. Para el NCAP se recomienda una plataforma que pueda ser portátil, como podría ser una Raspberry, para de esta manera poder implementar el sistema sin necesidad del uso de PC. Esto nos permite determinar que es posible desarrollar un módulo para sensores inteligentes que permita validar el protocolo 21451 de la IEEE utilizando un software libre para el desarrollo de las interfaces y una tarjeta con microcontrolador reprogramable integrado para el módulo que controlar los distintos transductores.

La conversión de un sensor genérico a un sensor inteligente no es laboriosa, ni requiere de un tiempo excesivo. Es posible grabar la hoja de datos de cualquier sensor genérico en una memoria de pequeñas dimensiones, y se podría instalar junto con el sensor o en cualquier punto del cable. Además, cabe mencionar que la memoria usada puede ser de muy bajo costo.

Además, si no es posible utilizar una memoria EEPROM, hay que tomar en consideración que la norma IEEE 21451.4 contempla la posibilidad de los “TEDS

5.2. RECOMENDACIONES

Virtuales” los cuales también pueden ser descargados desde internet.

Apéndice A

Tablas de datos para valores de los TEDS

Este apartado contiene las tablas de valores que se utilizan para el desarrollo de los TEDS.

Tabla A.1: Valores para datos Chr5.

		lsb							
		000	001	010	011	100	101	110	111
msb	00	Espacio	A	B	C	D	E	F	G
	001	H	I	J	K	L	M	N	O
	010	P	Q	R	S	T	U	V	W
	011	X	Y	Z	,	.	/	-	@

Tabla A.2: Números de identificación de los fabricantes.

ID	Nombre	ID	Nombre	ID	Nombre
17	Bruel & Kjaer	33	RION CO.	49	Sensorex
18	The Modal Shop, Inc.	34	Eensors	50	Waycon
19	Kistler Instrument Corporation	35	Dytran Instrument	51	AVL List GmbH
20	Macro Sensors	36	SincoTec	52	Pressure Systems
21	Wilcoxon Research	37	Lebow	53	ONO SOKKI
22	Endevco Corporation	38	Celesco	54	imc MeBsysteme GmbH
23	PCB Piezotronics, Inc.	39	Weed Instrument	55	Futek Advanced Sensor Technology, Inc.
24	G.R.A.S. Sound & Vibration	40	SensoTec	56	Knick Elektronische Messgeraete GmbH Co. KG
25	Larson Davis	41	Watlow	57	Lorenz Messtechnik GmbH
26	Key Transducers, Inc.	42	Transducer Techniques	58	Kyowa Electronic Instruments Co., Ltd
27	Oceana Sensor Technologies	43	Measurement Specialties	59	Interface, Inc.
28	Entek IRD	44	Fuji Ceramics	60	Sherborne Sensors Limited
29	Scitefair International, Inc.	45	Telemonitor	61	Metra Mess-und Frequenztechnik Radebeul
30	National Instruments	46	Lion Precision	62	Ametek Denmark A/S
31	HBM	47	LEM	63	Mettler-Toledo GmbH Process Analytics
32	TEAC	48	RDP Electronics	64	MTS Systems Corp.

Tabla A.3: Números de identificación de los fabricantes (continuación).

ID	Nombre	ID	Nombre	ID	Nombre
65	Viatran Corporation	82	Studio Technologies, Inc.	98	Proctor & Chester Measurements Ltd
66	Sinocera Piezotronics, Inc.	83	Dewetron GmbH	99	HEAD acoustics GmbH
67	MSC Automotive Ltd.	84	Natec Sensors GmbH	100	LLC GlobalTest
68	VIP Sensors	85	Adcon Telemetry GmbH	101	L.S. Starrett Company
69	Sensy SA	86	LCM Systems Ltd	102	teccis LP
70	Druck Ltd.	87	SVANTEK Sp. z.o.o	103	YMC Piezotronics, Inc.
71	Tokyo Sokki Kenkyujo Co.	88	Streamline Automation, LLC	104	Sentek Dynamics
73	Connection Technology Center	89	NTI AG	105	IPETRONIK GmbH & Co.KG
74	BSWA Technology Co., Ltd	90	Ridgetop Group Inc.	106	Dewesoft d.o.o
75	Sensordata Technologies, Inc.	91	Fenda Technology Ltd	107	Tovey Engineering Inc
76	The Pinax Group, Inc.	92	A.S.T. Agewandte System Technik GmbH	108	Ashcroft Inc
77	Kulite Semiconductor Products Inc.	93	Monitran	1097	ETAS
78	Showa Measuring Instruments Co., Ltd	94	Test GmbH	1098	Baker-Calling, Inc.
79	CEA	95	ACO, Co., Ltd	1099	ME-Messsysteme GmbH
80	Microtech Gefell GmbH	96	B&W Sensing Technologies	1100	Nakamura Mfg.Co., Ltd
81	Aikoh Engineering	97	ETH-messtechnik gmbh	1101	yurbuds Sport Earphones

Tabla A.4: Números de identificación de los fabricantes (continuación).

ID	Nombre	ID	Nombre	ID	Nombre
1103	LMS	1119	SIMTEC Elektronik GmbH	10874	STS Sensor Technik Sirmach AG
1104	UNIPULSE Corporation	1120	DJB Instruments	12385	TrigasDM GmbH
1105	EnvyLogic Co., Ltd.	2524	Prüftechnik Dieter Busch AG	14908	Flintec
1106	Dynamic Systems Inc	3123	Humanetics Innovative Solutions, Inc	15095	Soundtec GmbH
1107	DongHua Testing Technology Co., Ltd	3528	HAINBUCH GmbH SPANNENDE TECHNIK	15913	VALCOM CO., LTD
1108	CSM GmbH	5346	MinebeaMitsumi Inc.	16284	ZETLAB Company
1109	General Motors	5688	Duerr Somac GmbH		
1110	Druck- und Durchflussmessung	6457	CHENGDU CAIC ELECTRONICS CO., LTD		
1111	DPA Microphones A/S	6644	Siemens AG, MO MLT BG		
1112	Veridt, Inc	7382	Druck- und Durchflussmessung		
1113	Valens Semiconductor Ltd.	9588	Stankowitz Test Equipment GmbH		
1115	MICRO-EPSILON MESSTECHNIK GmbH & Co. KG	10156	mg-sensor GmbH		
1116	acam-messelectronic gmbh	10700	Hangzhou Aihua Instruments Co., Ltd		
1117	SKC Audio	10799	SENTER TECHNOLOGIES		

Tabla A.5: Tipo de datos usados en las plantillas estándar de la IEEE.

Tipo de dato	Descripción
UNINT	Entero sin signo
Chr5	Carácter de 5 bits
ASCII	ASCII estándar de 7 bits
<i>Date</i>	Número de días desde enero 1, 1998
<i>Single</i>	Punto flotante de precisión simple

Bibliografía

- [1] ARDUINO. <https://store.arduino.cc/usa/mega-2560-r3>.
- [2] ARDUINO. <https://www.arduino.cc/en/Reference/EEPROM>.
- [3] ARDUINO. <https://www.arduino.cc/reference/en/language/functions/communication/serial/read/>.
- [4] ARENY, R. P. *Sensores y acondicionadores de señal*, 4 ed. Marcombo boixareu editores, 2005.
- [5] BANNER. <https://www.bannerengineering.com/mx/es/company/expert-insights/io-link.html>, 27, Febrero de 2017.
- [6] CHEN, G., HUANG, J., AND LIN, R. Network interface UPnP mechanism of IEEE 1451 smart sensor., 2017.
- [7] FRADEN, J. *Handbook of modern sensors, physics, designs, and applications*, 3 ed. Springer, 2004.

BIBLIOGRAFÍA

- [8] GARCIA, J. A. S. Desarrollo de un sistema de adquisición de datos inteligente conforme a la norma IEEE std. 1451.4, 2013.
- [9] HUFNAGEL, P. The iee standard 1451.4 manufacturer id.
- [10] IEEE INSTRUMENTATION AND MEASUREMENT MAGAZINE.
- [11] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. Information technology -smart transducer interface for sensors and actuators- part 1: Network capable application processor (NCAP) information model, 2010.
- [12] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. Information technology -smart transducer interface for sensors and actuators- part 2: Transducer to microprocessor communication protocols and transducer electronic data sheet (TEDS) formats, 2010.
- [13] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. Information technology -smart transducer interface for sensors and actuators- part 4: Mixed-mode protocols and transducer electronic data sheet (TEDS) formats, 2010.
- [14] INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE taxonomy version, 2014.
- [15] JEVTIC, N., AND DRNDAREVIC, V. Development of smart transducers compliant with the IEEE 1451.4 standard.

BIBLIOGRAFÍA

- [16] LEE, K. B., AND SONG, E. Y. Object-oriented application framework for IEEE 1451.1 standard., 2005.
- [17] LOPEZ, C. <http://medicioni2007.blogspot.com>, 16, Junio de 2007.
- [18] MARK, J., AND HUFNAGEL, P. The IEEE 1451.4 Standard for smart transducers.
- [19] MICROSOFT NEWS. <https://www.msn.com/es-cl/noticias/microsoftstore/%C3%BAqu%C3%A9-es-y-para-qu%C3%A9-sirve-visual-studio-2017/ar-AAAnLZL9>.
- [20] NATIONAL INSTRUMENTS. An overview of IEEE 1451.4 transducer electronic data sheets (TEDS), 2016.
- [21] RAMÍREZ, D. S. G., LÓPEZ, M. J. L., AND DÍAZ, A. E. Válvula inteligente actuadora para sistemas hidráulicos., 2015.
- [22] ROSSI, S. R., SANTELLÁN, P. E., PRADO, T. A., AND DOS SANTOS FILHO, T. A. Sistema IEEE 1451.2 basado en microcontrolador y software JavaTEDS, 2007.
- [23] SONG, E. Y., AND LEE, K. B. Stws: A unified web service for IEEE 1451 smart transducers., 2008.
- [24] TSENG, C.-W., LIAO, Y.-T., AND HUANG, C.-H. Adding IEEE 1451 transducer capability to epcglobal information service, 2014.

BIBLIOGRAFÍA

- [25] WALL, D. R. W., AND HUSKA, A. Design platform for plug-and-play IEEE 1451 traffic signal.
- [26] WOBSCHELL, D., AND POH, W. S. A smart RTD temperature sensor with a prototype IEEE 1451.2 internet interface.