



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**IMPLEMENTACIÓN Y PRUEBAS DEL ESTÁNDAR DE COMUNICACIÓN DE
SERVICIOS DE DISTRIBUCIÓN DE DATOS PARA APLICACIONES DE MICRO-
REDES GESTIONADOS POR COMPUTADORES DE PLACA ÚNICA**

**TESIS PARA OPTAR AL GRADO DE MAGISTER EN INGENIERÍA DE REDES DE
COMUNICACIONES**

DANIEL ALEJANDRO ORELLANA GONZÁLEZ

**PROFESOR GUÍA:
CLAUDIO ESTÉVEZ MONTERO**

**MIEMBROS DE LA COMISIÓN:
GUILLERMO JIMÉNEZ ESTÉVEZ
SANDRA CÉSPEDES UMAÑA**

**SANTIAGO DE CHILE
2015**

RESUMEN DE TESIS PARA OPTAR AL TITULO
DE: Magister en Ingeniería de redes de comunicaciones.
Por: Daniel Alejandro Orellana González
Fecha: 24/09/2015
Profesor Guía: Claudio Estévez Montero

IMPLEMENTACIÓN Y PRUEBAS DEL ESTÁNDAR DE COMUNICACIÓN DE SERVICIOS DE DISTRIBUCIÓN DE DATOS PARA APLICACIONES DE MICRO- REDES GESTIONADOS POR COMPUTADORES DE PLACA ÚNICA

Las sociedades modernas son cada vez más dependientes de los suministros de electricidad confiable y segura que aportan al crecimiento económico y la prosperidad de la comunidad. En Chile y en el resto del mundo se piensa contar con varias micro-redes que pueden estar aisladas o pueden apoyar la labor de las plantas eléctricas, llevando a cabo la generación y administración eficiente de energía para determinadas zonas y/o poblaciones distantes. Para la construcción de tales redes es necesario poseer elementos de almacenamiento como baterías y elementos de generación de energía renovables, ya que estos desaceleran el consumo de combustibles fósiles.

Para un control eficiente de los elementos de la micro-red existe un considerable número de datos que se deben manejar, tabular y calcular, estos son una herramienta clave en el momento de tomar distintas acciones de generación, almacenamiento o distribución de la energía eléctrica en la red eléctrica. Para el buen manejo de parámetros en la micro-red existen estándares que permiten tener a la mano los datos importantes en tiempo real, ofreciendo velocidad, confiabilidad, interconectividad y un sin fin de buenas características para que de este modo se transporten y gestionen los datos de la mejor manera posible. Uno de los estándares usados en la actualidad que cumple con estas características es el servicio de distribución de datos (DDS, Data Distribution Service) de Object Management Group (OMG). La implementación de este estándar involucra equipos de comunicación configurados con DDS para tener un intercambio de datos entre los distintos elementos de la red.

Para el desarrollo de este proyecto se utilizará la versión de software DDS que ofrece la empresa RTI bajo una licencia académica gratuita de un año. Si se desea llevarlo a aplicaciones comerciales la licencia como tal tiene costo. El estándar ha sido implementado a nivel mundial, ya que es rápido, confiable y se pueden configurar de distintas formas para lograr la mayor eficiencia en el intercambio de datos de control y medición. Debido a las buenas características de este estándar, en esta oportunidad se implementará la instalación, configuración y pruebas de este en una red de comunicación compuesta por Single-board Computers (SBC), que son computadores económicos, rápidos, confiables y de fácil adquisición en el mercado. Una vez configurado el estándar de comunicación DDS se pondrá a prueba con la transmisión de la frecuencia de línea, una variable importante en el control y manejo de las micro-redes. Esta prueba de concepto demuestra que se puede recolectar cualquier parámetro de importancia y distribuirla por toda la micro-red.

Dedicatoria.

Dedico esta tesis a DIOS y a la Virgen María Auxiliadora ya que siempre me guiaron por el buen camino. A mis padres quienes me dieron vida, educación, apoyo incondicional y consejos para saber llevar esta maestría de la mejor manera. A mis profesores, compañeros y amigos, por contagiarme de esas ganas de superación en todo el transcurso de este programa de estudio y en este proyecto.

TABLA DE CONTENIDO

1	CAPITULO 1 INTRODUCCIÓN.....	8
1.1	Motivación.....	8
1.2	Problemática.....	8
1.3	Objetivo general.....	8
1.3.1	Objetivos específicos.....	9
1.3.2	Alcance.....	9
1.4	El Servicio Básico de la Energía Eléctrica en la Actualidad.....	9
1.5	Smart Grids.....	10
1.6	Micro-redes.....	12
1.6.1	Los elementos de una micro-red.....	13
1.6.2	La comunicación en la micro-red.....	15
2	CAPITULO 2 ESTADO DEL ARTE.....	17
2.1	Artículos.....	17
2.2	Principales estándares de comunicación para micro-redes.....	19
2.2.1	Ethernet.....	19
2.2.2	TCP/IP.....	20
2.2.3	IEC61850.....	21
2.2.4	Data Distribution Service.....	22
2.3	Sistemas de comunicación usados por empresas del mercado.....	23
2.4	Mejores sistemas de comunicación en micro-redes.....	25
3	CAPITULO 3 ESTANDAR DE COMUNICACION DDS PARA MICRO-RED.....	28
3.1	Bases de la teoría del estándar DDS.....	28
3.1.1	Metodología de comunicación DDS.....	29
3.1.2	Capas DDS.....	30
3.1.3	Calidades de servicio en DDS.....	33
3.2	DDS para solucionar retos SCADA.....	34
3.2.1	Soluciones en Datos.....	34
3.2.2	Soluciones en Redes.....	35
3.2.3	Solución en Integridad de datos.....	35
3.2.4	DDS en la Industria.....	37

3.2.5	DDS como estándar de comunicación en micro-redes.....	37
4	CAPITULO 4 PRUEBA DE CONCEPTO DE MICRO-RED CON DDS.....	39
4.1	DDS + Raspberry Pi	39
4.1.1	Instalación de DDS en el computador	40
4.1.2	Instalación de DDS en Raspberry Pi	40
4.1.3	Código C/C++ del cálculo y transmisión de la frecuencia en Raspberry Pi	40
4.2	Red DDS para pruebas de comunicación.....	40
4.2.1	Adquisición de datos de frecuencia de la red eléctrica.....	41
4.2.2	Programa C++ en Raspberry Pi para la recolección de datos de frecuencia.	43
4.2.3	Configuración de DDS para enviar datos de frecuencia.....	44
4.2.4	Consideraciones importantes en la red.	47
5	CAPITULO 5 RESULTADOS Y DISCUSIÓN.....	49
5.1	Pruebas y resultados del programa que calcula la frecuencia.....	49
5.2	Pruebas y resultados del envío de datos simples por DDS	52
5.3	Pruebas y resultados del envío de una sola frecuencia por DDS.....	53
5.4	Pruebas en el manejo de Tópicos.....	54
5.5	Pruebas y resultados del envío de dos diferentes frecuencias por DDS.	56
5.6	Análisis de paquetes DDS de frecuencia.	57
5.6.1	Paquetes recogidos desde un publicador o suscriptor	58
5.6.2	Paquetes recogidos desde una Pc sin ser publicador ni suscriptor.	59
5.7	Comparación del estándar DDS (UDP) Vs. Aplicaciones (TCP).....	60
6	CONCLUSIONES Y TRABAJO FUTURO.....	62
6.1	Conclusiones	62
6.2	Trabajo futuro	63
7	BIBLIOGRAFÍA.....	64
	ANEXO A	67
	TUTORIAL PARA LA INSTALACION DE DDS DE RTI EN EL COMPUTADOR.....	67
	ANEXO B	81
	TUTORIAL PARA LA INSTALACIÓN DE DDS DE RTI EN RASPBERRY PI.....	81
	ANEXO C	89
	CÓDIGO C/C++ DEL CÁLCULO Y TRANSMISIÓN DE LA FRECUENCIA EN RASPBERRY PI.....	89

Índice de Tablas

Tabla 1: Documentos científicos sobre los últimos avances de la tecnología en micro-redes.	18
Tabla 2: Empresas que ofrecen la implementación y comunicación de la micro-red.	24
Tabla 3: Características de los sistemas de comunicación para micro-redes.	27
Tabla 4: Comparación de resultados de medida entre DDS y TCP/IP.	60

Índice de Figuras

Figura 1: Definición de Smart Grid. Fuente: www.simplydecoded.com	11
Figura 2: Estructura de una micro-red. Fuente: CEDETEL	13
Figura 3: Componentes de una micro-red. Fuente: www.ecsblog.org	14
Figura 4: Comunicación en la micro-red Fuente: www.zeitgeistlab.ca	16
Figura 5: Modelo de referencia OSI y el conjunto de protocolos TCP/IP. Fuente: www.securenet2002.tripod.com	21
Figura 6: Esquema básico de comunicación DDS Fuente: www.prismtech.com	28
Figura 7: Comparación de DDS con el Modelo OSI. Fuente: Propia	30
Figura 8: Ubicación de cada capa del sistema de comunicación DDS. Fuente: www.cs.wustl.edu/~schmidt/cs395/DDS-intro.ppt	30
Figura 9: Elementos de la capa DCPS. Fuente: www.prismtech.com	32
Figura 10: Calidades de Servicio presentes en componentes DDS. Fuente: www.prismtech.com	34
Figura 11: Participantes, nodos y dominios de DDS. Fuente: www.prismtech.com	36
Figura 12: Raspberry Pi o Single Board Computer (SBC). Fuente: www.raspberrypi.org	39
Figura 13: Topología de la red DDS para pruebas y mediciones. Fuente: Propia.	41

Figura 14: Esquema electrónico de la placa intermedia para la recolección de datos de frecuencia. Fuente: Propia.....	42
Figura 15: Esquema electrónico para armar el circuito de recolección de datos de frecuencia. Fuente: Propia.....	43
Figura 16: Periodo de la señal T y cálculo de la frecuencia. Fuente: www.fisicapractica.com	44
Figura 17: Parte del código original del programa HelloPublisher.c de DDS. Fuente: Propia	45
Figura 18: Parte del código modificado del programa HelloPublisher.c de DDS. Fuente: Propia	45
Figura 19: Parte del código en donde está el tópico asociado. Fuente: Propia.....	46
Figura 20: Directorio de archivos de Hello_simple3. Fuente: Propia.	46
Figura 21: Elementos de la red y tópicos asociados. Fuente: Propia.	48
Figura 22: Datos de frecuencia del generador de funciones y del programa C++. Fuente: Propia	49
Figura 23: Calculo de la varianza, desviación estándar y error estándar de la media. Fuente: Propia.....	50
Figura 24: Histograma de los valores de frecuencia obtenidos desde la Raspberry Pi.	51
Figura 25: Pruebas de transmisión de texto simple con DDS. Fuente: Propia.	52
Figura 26: Prueba de trasmisión de datos reales de frecuencia con DDS. Fuente: Propia.....	53
Figura 27: Publicadores con diferente tópico en DDS. Fuente: Propia.....	55
Figura 28: Dos publicadores y dos suscriptores con diferentes. Fuente: Propia.	56
Figura 29: Envío y recepción por DDS de dos frecuencias. Fuente: Propia	57
Figura 30: Paquete 100 de transmisión de frecuencia de 60Hz por DDS visto desde un suscriptor. Fuente: Propia.....	58
Figura 31: Paquete 100 de transmisión de frecuencia de 60Hz por DDS visto desde una Pc cualquiera. Fuente: Propia.	59

1 CAPITULO 1 INTRODUCCIÓN

El primer capítulo comienza planteando la motivación para la elaboración de este proyecto, así como también la problemática y objetivos. Luego se continúa con una breve descripción sobre la energía eléctrica, la importancia de su uso en estos días y a futuro. Además se habla brevemente de los métodos de generación de energía, fuentes de energía renovable y la descripción del concepto de una micro-red con sus respectivos detalles, como su funcionamiento, principales características, sus elementos y la comunicación efectiva dentro de esta.

1.1 Motivación

La causa por la cual se desarrolla este proyecto es poder implementar y probar un estándar de intercambio de datos y proponer la aplicación de este en la construcción de futuras micro-redes en Chile y en el mundo. Además este proyecto pretende dar una posible solución en la parte de comunicación a un proyecto de I + D del área de energía en el cual se buscaba un sistema de comunicación y control para los elementos de una micro-red instalada en Huatacondo - Chile. Es por este motivo que nació la idea de buscar la mejor forma de comunicación para tales componentes e implementar un laboratorio para las pruebas respectivas de intercambio de datos reales.

1.2 Problemática

En la actualidad y en varias partes del mundo, la instalación de pequeñas y medianas micro-redes para satisfacer las necesidades de servicios eléctricos está creciendo considerablemente. Un artículo de la revista de Stanford publicado el 9 de agosto del presente [1] dice que existen planes en algunos países para aumentar a un 80 – 85% la capacidad de generación de energía eléctrica con fuentes de generación como agua, viento y luz solar. Esto es viable por dos principales razones: para aprovechar las mejoras tecnológicas de los últimos años y para disminuir la contaminación por el consumo de combustibles fósiles para la producción de energía. El problema es que en el mercado existen diferentes soluciones para la comunicación de los elementos de una micro-red las cuales se ofrecen a costos elevados, tratando de monopolizar toda la red. Además algunas de estas soluciones empresariales implementan estándares con tecnología de años anteriores. Adicionalmente, se argumenta que la administración eficiente de energía a pequeña escala debe consistir de una comunicación distribuida, en contraste con un sistema centralizado. Centralizar el control, por ejemplo a través de un servidor, puede tener consecuencias muy perjudiciales ya que si un controlador central de comunicación se desconecta toda la red falla. La distribución de la inteligencia en la red de comunicaciones crea un ambiente inherentemente más confiable, rápido, escalable y su mantención y reparación es más económica.

1.3 Objetivo general

El objetivo de este trabajo es implementar un sistema distribuido para habilitar el desarrollo de aplicaciones de micro-redes usando un estándar moderno de comunicación denominado DDS por medio del monitoreo continuo de un parámetro predefinido, en este caso frecuencia eléctrica, y la

programación de una plataforma que consiste en un computador de placa única como elemento de gestión.

1.3.1 Objetivos específicos

- Analizar el estado del arte de la academia y de la industria de las micro-redes
- Realizar el análisis de los actuales estándares de comunicación para micro-redes.
- Investigar acerca del estándar de mejores prestaciones DDS.
- Instalar y configurar el estándar DDS en equipos de red.
- Diseñar una red de comunicación basada en el estándar DDS que sirva como prueba de concepto de una micro-red.
- Realizar pruebas y análisis de paquetes.

1.3.2 Alcance

El alcance del presente proyecto de tesis se centra en el estudio y aplicación de una prueba de concepto de un sistema descentralizado para aplicaciones relacionadas a micro-redes que se basa en el estándar DDS. También se realiza una revisión de las más recientes investigaciones y desarrollos de estándares de comunicación para micro-redes sin profundizar en cada una de estas. En la revisión de la teoría del estándar se presenta tan solo lo relevante a DDS para tener las bases, descubrir las cualidades y ventajas. Posteriormente se diseña y desarrolla una red de laboratorio que involucra equipos de red y de comunicación bajo el estándar DDS. Finalmente se pretende transmitir por DDS un parámetro importante de medición en micro-redes que es la frecuencia eléctrica para observar el comportamiento de la red, realizar pruebas de la recolección y análisis de paquetes en la red en cuestión. La frecuencia eléctrica será calculada midiendo el tiempo de un periodo con señales sinusoidales puras sin contenido armónico. No es alcance de esta investigación llegar a definir la forma en la que se conectarán los elementos de control de la micro-red ni definir un software de control o gestión.

1.4 El Servicio Básico de la Energía Eléctrica en la Actualidad

Las ciudades hoy en día se enfrentan a un gran desafío: ser sostenibles tanto en factores económicos como medioambientales a largo plazo. Para el desarrollo de cualquier región, las ciudades y capitales se han convertido en ejes principales del crecimiento económico, innovación, progreso social, cultura, conocimiento y diversidad. De esta forma se debe considerar la calidad de los servicios básicos, como por ejemplo la energía eléctrica, como uno de los aspectos de atracción más importantes. La concentración urbana en sociedad conlleva a que las ciudades sean responsables de los grandes retos de sostenibilidad de la región teniendo en cuenta que las ciudades consumen más de dos tercios de la energía mundial y representan el 70% de las emisiones globales de CO₂ [2].

Un reciente informe publicado por la ONU da a conocer que: si mantenemos el actual modelo de consumo, en el año 2030 las necesidades de la sociedad habrán crecido exponencialmente y el mundo necesitará el 50% más de comida, el 45% más de energía y el 30% más de agua [2]. En

otro estudio realizado por, “Recursos Energéticos y Crisis” (Carles Riba, UPC, 2011), se estima que las reservas de energía no renovables (carbón, petróleo, gas y uranio) se agotarán entre el 2060 y 2070, estas medidas involucran un incremento de consumo y población. Con estas publicaciones se puede decir que la unión de los efectos del crecimiento demográfico, cambio climático y contaminación, pueden causar impactos negativos realmente considerables en la calidad de vida y en la estabilidad económica y social [3].

Para enfrentar estos retos existen en la actualidad diferentes tecnologías que facilitan la generación más eficiente y mejor aprovechamiento de cada uno de los recursos. Uno de los recursos considerado básico para la sociedad es la energía eléctrica que está presente en la mayoría de hogares del mundo para satisfacer las necesidades que tienen las personas como iluminación y funcionamiento de aparatos eléctricos en general. Con el paso de los años la energía eléctrica ha tenido cada vez más importancia, ya que la tecnología avanza y por lo general los nuevos y sofisticados aparatos depende del consumo energético para su funcionamiento. En resumen se puede decir que la mayoría de nuevos artefactos tecnológicos que lanza el mercado tiende a la utilización de la energía eléctrica como medio de alimentación y es requisito indispensable para el buen funcionamiento de cada dispositivo.

La red eléctrica actual se construyó para permitir a las empresas de servicios suministrar la energía eléctrica a los hogares de los consumidores y poder facturar una vez al mes el servicio prestado. Esta interacción unidireccional con limitaciones dificulta la capacidad de la red para dar respuesta a la creciente demanda de energía del siglo 21 sujeta a un continuo cambio. Debido a esto en los últimos años nace un nuevo concepto denominado Smart Grid que incorpora un dialogo bidireccional con la característica de poseer un intercambio de electricidad e información entre la empresa suministradora y sus clientes. Se trata de una red de desarrollo de comunicaciones, controles, ordenadores, sistemas de automatización, nuevas tecnologías y herramientas que interaccionan entre sí con el fin de hacer a la red más eficiente, más fiable, más segura y más respetuosa con el medio ambiente. La red inteligente permite la interacción de nuevas tecnologías como la producción de energía eólica, solar así como también la carga de vehículos eléctricos. Cabe destacar que la Smart Grid tiene la posibilidad de trabajar conectada o no a la red principal de generación eléctrica. Gracias a todas estas cualidades las empresas suministradoras de servicios eléctricos podrán comunicarse mejor con sus clientes para optimizar la gestión de sus necesidades en materia de energía eléctrica.

A continuación se describirá mejor lo que son las Smart Grids y micro-redes ya que estas dos soluciones son en la actualidad las nuevas formas de gestión de la generación eléctrica.

1.5 Smart Grids

Los sistemas de generación como la energía eólica y la energía solar destacados anteriormente son muy variables por lo que es muy necesario disponer de un sistema de control más sofisticado como son las redes eléctricas inteligentes (REI) para facilitar las conexiones de otras fuentes a la red. La potencia de células fotovoltaicas y la de los parques eólicos han sido significativa, por lo que ha puesto en duda el autoritario poder de las centrales eléctricas grandes y centralizadas. La

rápida caída de los costos apuntan a un cambio importante en la topología de red, pasando de ser centralizada a uno altamente distribuido, con el propósito de poder ser generada y consumida de acuerdo al tamaño de la red.

La red eléctrica inteligente (REI) conocida también como Smart Grid se puede conceptualizar como la unión dinámica de los avances tecnológicos de la información y comunicación (TIC) y los desarrollos modernos en ingeniería eléctrica y electrónica dentro de todo el campo de la producción de energía, en esto se incluyen las partes que constituyen la red como generación, transmisión, distribución y comercialización con la inclusión adicional de energías renovables, todo esto manejado conjuntamente desde un solo sistema de gestión inteligente. Esto permite que las áreas de coordinación como protección, control, instrumentación, medida, calidad y administración sean agrupadas con el principal objetivo de realizar un uso eficiente y racional de la energía, de esta manera se asegura un equilibrio coordinado entre la oferta y la demanda entre productores y consumidores.

La tecnología cada vez mas involucrada en la sociedad, juega un papel primordial en el desarrollo de las Smart Grids por lo que contribuye en varios aspectos de forma positiva a cumplir el objetivo de un uso eficiente de energía. Así mismo las Smart Grids se caracterizan por la inclusión de energías renovables y muchas veces poseen la particularidad de que los usuarios no solo consumen sino que también producen electricidad a través de la misma red, con la instalación de generadores típicamente eólicos y solares en los hogares. Por tanto, el flujo de energía es ahora bidireccional, esto se da cuando se envía electricidad desde los proveedores a los consumidores y viceversa usando una tecnología digital para controlar las necesidades del consumidor. Esto ayuda a ahorrar energía, reducir costos e incrementar la usabilidad y transparencia.

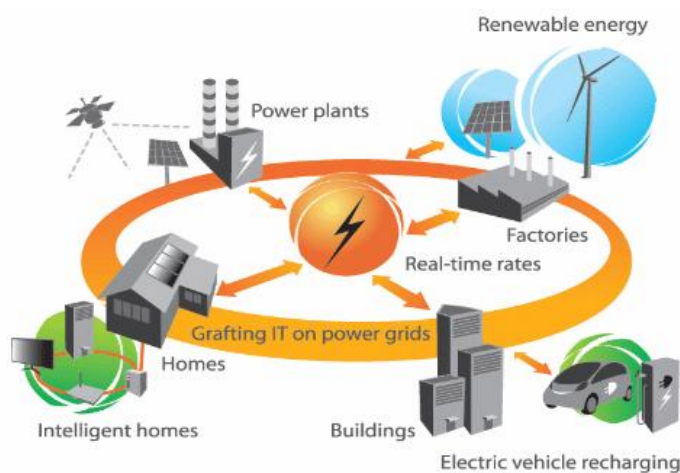


Figura 1: Definición de Smart Grid. Fuente: www.simplydecoded.com

La figura 1 muestra los diversos elementos con los que se compone una red inteligente como son los hogares inteligentes, vehículos eléctricos recargables, industrias y fábricas, también están las micro-redes con sus fuentes de energía renovables y las plantas de generación convencional. Además se puede ver la interconexión de estos elementos conformando un sistema en la cual los

componentes se comunican a través de una red de transporte de datos y un sistema gestor. Todo esto se realiza con la idea principal de mejorar la eficiencia tanto en la generación con el consumo de la energía eléctrica. Como se dijo anteriormente dentro de todos los elementos que conforma la Smart Grid se encuentra el trabajo de las micro-redes la cual se describirá en el siguiente punto.

El uso de fuentes de generación o producción de energía alternativos con recursos renovables que incluyen la energía eólica y la solar entre las más importantes forman parte de una Smart Grid y considerando que esta red inteligente ayuda a la eficiencia y fortalecimiento de la generación de un recurso indispensable como es la energía eléctrica se puede afirmar que una Smart Grid forma parte a su vez de una Smart City ya que buscan un fin común que es la sostenibilidad y eficiencia de los recursos para la sociedad.

1.6 Micro-redes

Desde muchos años la generación hidráulica de energía eléctrica ha sido en muchos países la principal fuente de electricidad aprovechando muchas veces el agua dulce para este proceso. En países en donde no existe tal recurso se ha optado por diferentes formas de generación eléctrica entre las cuales se encuentra el uso de combustibles fósiles generando impactos ambientales y sociales en el mundo contribuyendo de esta manera al calentamiento global. Lo anteriormente mencionado sumado a que los recursos hídricos que dependen fuertemente de los ciclos climáticos ha dado lugar a la búsqueda de nuevas estrategias de generación de energía a partir de recursos renovables no convencionales (viento, sol, mar y otros). Con los avances tecnológicos dados en los últimos años, las turbinas eólicas, paneles fotovoltaicos y concentradores solares han tenido un gran avance en su desarrollo, de esta manera el viento y el sol han sido considerados como opciones reales para la generación eléctrica aunque la variabilidad e incertidumbre de este tipo de recursos dificulta su alta penetración en los sistemas eléctricos de potencia convencionales.

Con el objetivo de alcanzar una elevada integración de energías renovables no convencionales y mejorar la eficiencia actual de los sistemas eléctricos de potencia convencionales de generación local para cada ciudad, en los últimos años se ha estado desarrollando una nueva forma de generación de energía eléctrica llamada Generación Distribuida, la cual se enfoca en el uso de generadores pequeños también conocidos como recursos energéticos distribuidos, conectados entre sí para formar una red eléctrica y satisfacer la demanda de los usuarios. Por lo tanto la Generación Distribuida sería la opción más acertada para explotar los recursos renovables distribuidos como el viento, el sol, pequeños recursos hídricos, biomasa, entre otros presentes en localidades geográficas.

Se espera que a futuro los actuales sistemas eléctricos, basados en pocas pero grandes centrales de generación, den paso a la incorporación de gran cantidad de pequeños generadores apoyados por fuentes de energías renovables no convencionales, como el sol y el viento. Sin embargo, un importante incremento de generadores distribuidos de pequeña escala produciría un desafío técnico relacionado con el control de estabilidad de todo el sistema debido a la característica unidireccional de las actuales redes de transporte de electricidad. Una manera prometedora de

explotar el emergente potencial de la Generación Distribuida sin afectar la estabilidad del sistema es tomando un nuevo enfoque sistémico que vea a los generadores y cargas como un sub-sistema o una “Micro-red” [4].

De manera formal, una micro-red se define como un sistema eléctrico de distribución que contiene cargas y recursos energéticos distribuidos, como generadores distribuidos, dispositivos de almacenamiento y cargas controladas, los cuales pueden ser operados de forma coordinada ya sea mientras está conectado a la red principal de potencia o aislado de ella [5]. En la actualidad esta tecnología ha ido evolucionando y se considera su aplicación en: ciudades pequeñas, edificios, industrias, comunidades asiladas, complejos comerciales y redes de emergencia. Adicionalmente esta aplicación se puede considerar en micro-redes domésticas, que en general cumplen con los mismos principios y que se las conoce como nano-redes.

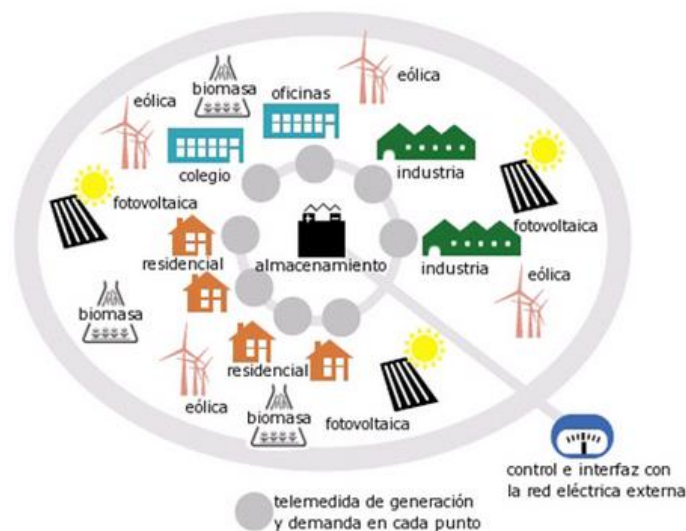


Figura 2: Estructura de una micro-red. Fuente: CEDETEL

En la figura 2 se pueden apreciar los elementos que conforman una micro-red entre los cuales se destacan los generadores de energía como: parques eólicos, paneles fotovoltaicos, biomasa, así como también cargas controladas como: colegios, industrias y residencias por citar algunas. Además en la figura se destaca un equipo de almacenamiento de energía que generalmente está formado por bancos de baterías, todos estos elementos a su vez son conectados entre sí para ser monitoreados y controlados por una estación central la cual dispone de un programa gestor de toda la micro-red. Cabe destacar que esta red será capaz de generar electricidad para el consumo de una determinada zona pudiendo o no ser conectada a redes eléctricas externas para su correcto funcionamiento.

1.6.1 Los elementos de una micro-red

Anteriormente se dijo que básicamente una micro-red es un sistema eléctrico de distribución que contiene cargas y recursos energéticos distribuidos en la que se presenta la interoperabilidad de todos los componentes de esta red con la interacción de las tecnologías de la información.

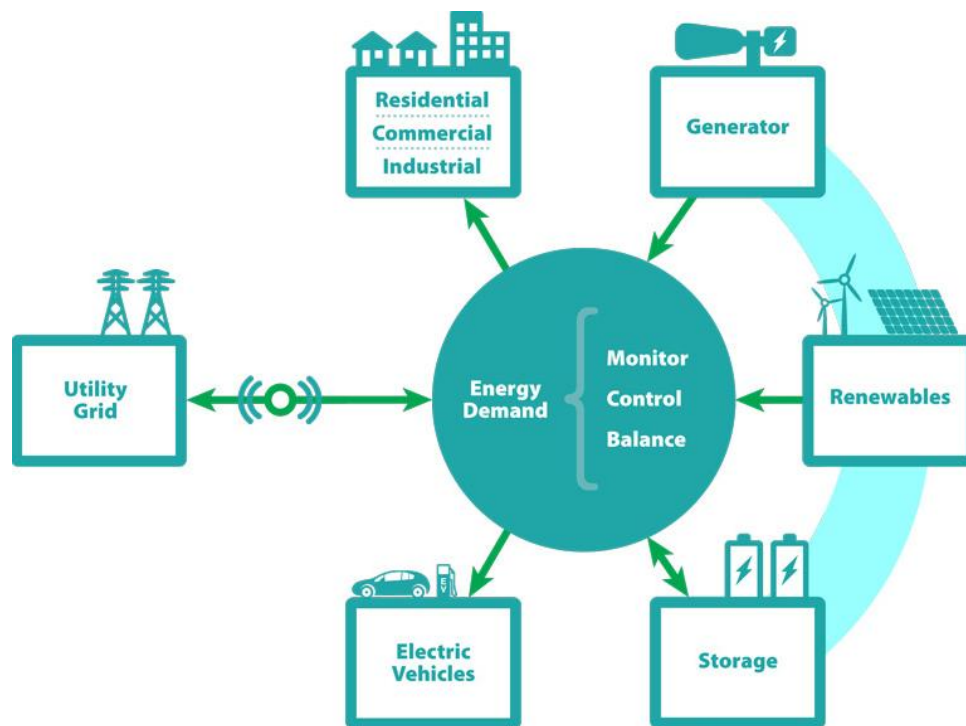


Figura 3: Componentes de una micro-red. Fuente: www.ecsblog.org

En esta oportunidad, la figura 3 indica de manera resumida y de forma agrupada los componentes que tiene una micro-red. En el grupo de almacenamiento o storage se encuentran los elementos que adquieren energía y la guarda para un posterior uso, dentro de estos elementos están principalmente las baterías o bancos de baterías, además estas dan sostenibilidad al sistema y la energía eléctrica fluirá en los dos sentidos desde la red de generación a las baterías o desde estas hacia la red de distribución. En el grupo de renovables o renewables están las plantas de generación de electricidad en forma no convencional, haciendo uso del sol se encuentra la planta fotovoltaica y aprovechando el viento está el parque eólico. El grupo de generación (“Generator”) hace referencia a generadores que ayudan a la producción de electricidad generalmente quemando combustibles fósiles los cuales contribuyen al calentamiento global lo cual hoy en día se trata de minimizar o desaparecer. La figura también muestra las cargas que debe tener la micro-red, estas consumirán la energía generada y almacenada por la red. Cabe destacar que las cargas también forman parte importante de esta red, las cuales se pueden controlar. Entre estas pueden destacar hogares, escuelas, hospitales, industrias y vehículos eléctricos

Dentro de una micro-red típicamente existe un sistema de gestión o manejo de los elementos que conforman la red los cuales pueden ser generadores, almacenadores o consumidores de energía que permite que toda la red este operando de manera estable y segura, logrando tener una red que sea confiable y que esté disponible a los usuarios y consumidores finales. Este sistema además brinda la posibilidad de monitorear, controlar y balancear la energía y las cargas presentes en la red, considerando esto, la micro-red tiene la posibilidad de configurar su sistema de gestión y permitir que la red trabaje aislada o conectada a la red de distribución convencional.

1.6.2 La comunicación en la micro-red.

Para lograr gran flexibilidad en las operaciones de un micro-red se necesita un sistema integrado y jerárquico¹ que permita gestionar y controlar la red de manera correcta. Una amplia gama de tecnologías de la comunicación, medios de comunicación, protocolos, sensores y algoritmos están disponibles para aplicaciones micro-red en el mundo. Los requisitos de capacidad de comunicación difieren de una red a otra, de su tamaño y de su forma de implementación y construcción. Otra característica relevante que debe tener el sistema de control es la integración eficiente de los recursos de energía distribuida (DERs, Distributed Energy Resources), ya que una micro-red está constituida por un pequeño sistema en el que generadores distribuidos y cargas eléctricas se colocan juntos y se controlan de una manera eficiente y completa [5]. Esto permite la utilización del parámetro de nivel de carga medido en la red para el control de flujo de potencia a ser entregado a los usuarios, constituyendo una operación eficiente y planificada de generación distribuida teniendo en cuenta la economía de la red y la eficiencia energética.

Una micro-red debe tener su propio sistema de gestión para apoyar las funciones de control necesarias y para autónomamente regular los flujos de energía[7], así como a participar en el mercado de la energía para el comercio de la electricidad [8].

Cuando se trata de explicar cómo se produce el intercambio de datos entre dos dispositivos de una micro-red se debe recordar los conceptos de arquitectura de comunicaciones y topología de red. Arquitectura de comunicaciones es una estructura organizada jerárquicamente que permite enviar y recibir datos entre niveles lógicos semejantes en distintas maquinas o terminales de la misma o distinta red, como por ejemplo niveles lógicos del modelo OSI. Por otro lado topología de red hace referencia a la forma en la que está diseñada la red y pueden ser de varios tipos desde descentralizada peer-to-peer hasta topologías centralizada maestro-esclavo cliente-servidor pasando por topologías tipo anillo, bus, árbol, etc. La configuración peer-to-peer, proporciona controles de frecuencia y voltajes rápidos y muy fiables localmente, sin embargo, este tipo de topología tiene limitaciones en lograr un alto rendimiento del sistema y aplicaciones avanzadas debido a la falta de comunicaciones [9].

A largo plazo, la necesidad de una estrecha supervisión del sistema y un mayor rendimiento, junto con el desarrollo de tecnologías de comunicación cada vez más avanzadas promoverán la integración de la comunicación con la micro-red. Los requisitos y capacidades de comunicación difieren entre una y otra variando las escalas físicas, las estrategias de control y funciones. Por ejemplo, el voltaje en tiempo real y el control de frecuencia requieren ser más confiables, más frecuentes y más rápidos en el intercambio de información que el despacho económico. El mayor problema está en cómo elegir la solución más confiable de rendimiento, que sea rentable y que cumpla con todos los requisitos y funciones de una micro-red. Los ingenieros en comunicación y sistemas de energía están trabajando en las soluciones. Con énfasis en la configuración de las comunicaciones, una red Wi-Fi es probada en una pequeña micro-red de energía gestionada desde un entorno de laboratorio en [10], mientras el estándar RS485 con la interfaz de LabVIEW

¹ Jerárquico: relacionado a la organización por grados de importancia

son considerada en [11] y comunicación en tiempo real mediante el estándar IEC61850 con Data Distribution Service (DDS) en [13]. Estos temas y otros serán tratados más adelante.

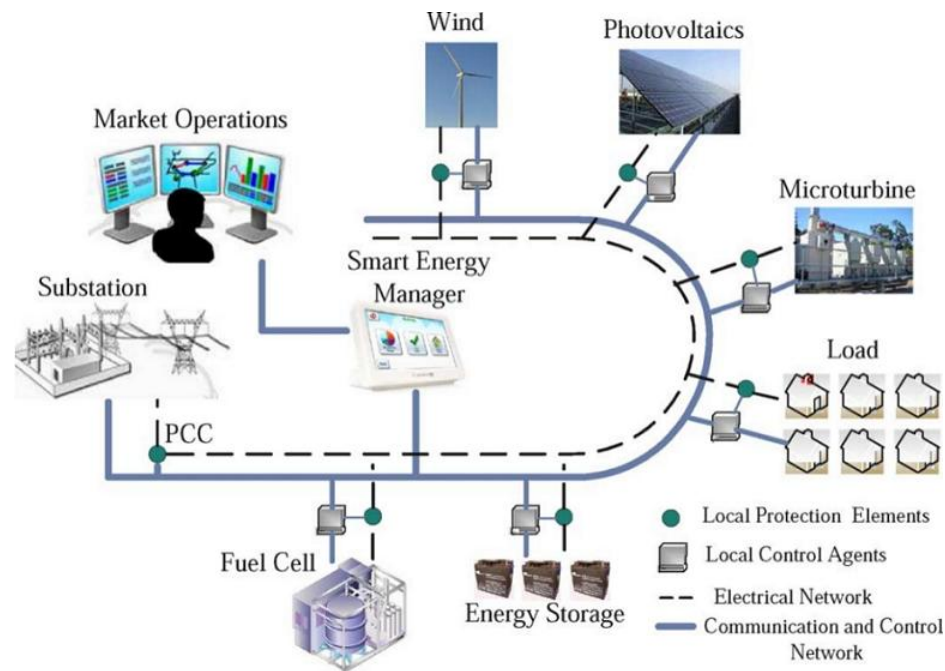


Figura 4: Comunicación en la micro-red Fuente: www.zeitgeistlab.ca

En la figura 4 se puede apreciar un esquema de comunicación utilizado para una micro-red en el cual se presentan dos redes, la primera es la red eléctrica por donde es transportada la energía la cual presenta dispositivos de protección y la segunda es una red de comunicación con una topología tipo bus por donde se da el intercambio de datos de información. También se observa que existen agentes o dispositivos de control local conectados a la red de comunicación los cuales contribuyen en la medición de los distintos parámetros eléctricos e intercambian datos en la red, además sirven de interfaz o mediador entre los distintos componentes de la micro-red y la propia red de comunicación. Muchas veces los agentes convierten las señales de datos entregados por elementos de la red a un formato común para que esta información pueda ser retransmitida por toda la red sin inconveniente.

La red que se muestra dispone adicionalmente de un centro de monitoreo y control para realizar diversas estadística, mediciones y observar el comportamiento de los elementos y su correcto funcionamiento. El centro de gestión de red puede o no disponer de un gestor inteligente de energía para no solo observar la estabilidad de la red sino además realizar un análisis del comportamiento de la red, pronóstico de generación y demanda para poder dar una mayor eficiencia en la generación y uso de la energía. Sin un apropiado centro de control, la micro-red no puede aprovechar las ventajas comparativas de contar con energías renovables. El mejoramiento de los actuales sistemas de control permitirá el desarrollo de las Redes Inteligentes del futuro.

2 CAPITULO 2 ESTADO DEL ARTE

Con los conceptos anteriormente mencionados sobre micro-redes y la comunicación entre sus componentes, es posible revisar las investigaciones que se han realizado en los últimos años en este tema. Se dispondrá principalmente de artículos de la academia para repasar los trabajos realizados en micro-redes y la aplicación de distintos estándares y protocolos usados para el intercambio de datos y la comunicación entre los elementos de esta red. Existen básicamente tres nuevos términos que son muy usados en este ámbito los cuales se describen a continuación:

- **Interfaz:** Se define interfaz como todo puerto que nos permite enviar y recibir señales desde un componente a otro, teniendo entonces distintas formas de realizar este envío y dispuestas por las especificaciones técnicas de cada equipo, o bien mediante el establecimiento de distintos estándares que permiten la comunicación.
- **Protocolo:** Un protocolo de comunicaciones es un conjunto de reglas y normas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellos para transmitir información. Se trata de reglas o el estándar que definen la sintaxis, semántica y sincronización de la comunicación así como posibles métodos de corrección de errores de datos.
- **Plataforma:** Software de comunicación o programas que intercambian mensajes e información por medio de los elementos de la micro-red, permitiendo tener una forma fácil y rápida para el intercambio de datos entre el usuario y los dispositivos. Los softwares de comunicación ofrecen soluciones cómodas para las necesidades de comunicación.

2.1 Artículos

Se identificaron varias investigaciones relacionadas con los sistemas de comunicación en micro-redes. Los artículos más relevantes, incluidos en esta revisión, son:

- Communication platform for energy management system in a master-slave control structure Microgrid [12].
- Mapping of IEC 61850 to Data Distribute Service for Digital Substation Communication [13]
- Communication technologies for BCIT Smart Microgrid[14].
- Design of Arduino-based Communication Agent for Rural Indian Microgrids [15].
- The smart microgrid pilot project of the University of Genoa: Power and communication architectures [16].

A continuación se presenta la Tabla 1 comparativa de los artículos seleccionados. En ella se comparan los objetivos, las características de comunicación, la interfaz, el protocolo, la plataforma y las velocidades de cada sistema.

Ref.	[12]	[13]	[14]	[15]	[16]
	Año	2012	2013	2012	2014
País	China	China	Canadá	India	Italia
Resumen	Construcción de una nueva plataforma de comunicación para la micro-red con estrategia maestro – esclavo. (Modbus)	Propuesta de sistema de comunicación confiable y en tiempo real con el mapeo de IEC61850 usando Data Distribution Services) de Object Management Group.	Revisión de los sistemas de comunicación de extremo a extremo del control dentro de las micro-redes inteligentes.	Diseño de sistema de comunicación con las funciones y componentes de un Arduino y Módulo GSM asociados a implementar una micro-red.	Desarrollo y pruebas de control, dispositivos y componentes de aplicaciones Smart Polygeneration Microgrid (SPM).
Características Comunicación	Uso completo de las funciones de comunicación Digital Signal Processing (DSP).	IEC proporciona especificaciones para los parámetros de micro-redes. DDS permite la comunicación en tiempo real entre los elementos.	Evaluación de resultados iniciales de WiMAX y ZigBee como red de comunicación British Columbia Institute of Technology (BCIT).	Sistema multi-agente jerárquico (MAS). Datos se convierten en señales GPRS dentro del Arduino mediante un pseudocódigo.	Sistema de comunicación con dos switches (para redundancia) conectados a otros por medio de fibra óptica.
Interfaz	Interfaz de comunicación Serial Communication Interface (SCI) e interfaz Ethernet.	Sin información.	Interfaz de radio frecuencia	Arduino y módulo GSM	Fibra óptica de 10 a 100 Mbps para una conexión en anillo.
Protocolo o Estándar	Controller Area Network (CAN) - abierto entre el maestro y los dispositivos inversores esclavos.	DDS e IEC61850.	Protocolo de integración WiMAX y ZigBee. Para el intercambio de datos IEC 61850.	GSM/GPRS	Protocolo Ethernet TCP/IP.
Plataforma	Propio desarrollo de plataforma de comunicación múltiple que contiene una red Ethernet y una CAN bus dentro.	DDS-RTI. Data Distribution Services de la empresa RTI.	Sin información.	Paquetes de datos GPRS desde y hacia el servidor de la nube.	Decentralized Energy Management System (DEMS) de Siemens.
Velocidades	La velocidad de transmisión de la comunicación en serie 9600 bits/s.	Sin Información	La velocidad de datos de prueba fue 9Mbps, 50% para el DL y 50% para el UL.	Sin información	Sin información

Tabla 1: Documentos científicos sobre los últimos avances de la tecnología en micro-redes.

Estos trabajos relatan casos prácticos de diseño e implementación de sistemas de comunicaciones para micro-redes. Algunas de estas soluciones se enfocan en sistemas de comunicaciones cableadas, inalámbricas o una mezcla de ambas, tratando de que sean cada vez más confiables y veloces en lo que respecta a la medición y control de las variables operacionales de las micro-redes. En redes cableadas se utilizan protocolos de integración como Modbus y CAN en [12] y Ethernet TCP/IP en [16] entre los más importantes. DDS sobre Ethernet es una de las opciones que se ofrece en la actualidad ofreciendo comunicación en tiempo real, estandarización de elementos de la red por medio del IEC61850 y un modelo publicador- suscriptor en [14]. Uno de los puntos más nuevos que sobresale entre los materiales usados para la comunicación en micro-redes es la fibra óptica en [16] la cual ofrece sin duda las mayores resoluciones y velocidades de datos capaz de cubrir grandes distancias. La única limitante para la decisión de su implementación es el costo tanto de equipamiento como de instalación.

Gran variedad de micro-redes utilizan cables para la comunicación, pero además de estas se ha incorporado el desarrollo de estas redes con tecnología inalámbrica en [15] que funcionan muy bien en la mayoría de ambientes abiertos y despejados con un cierto margen de distancia para su buen desempeño. Esta transmisión inalámbrica están cubierta con algunos otros protocolos de integración como WiMAX definido por la IEEE 802.16 en [15], Wi-Fi definido por la IEEE 802.11 en [15], Zigbee definido por la IEEE 802.15 en [15] y la integración actual de tecnología GSM con un módulo Arduino en [16].

2.2 Principales estándares de comunicación para micro-redes.

Cuando se pretende explicar la metodología de comunicación y el manejo de datos en las micro-redes es necesario considerar que el intercambio de datos es un proceso el cual se rige por un conjunto de reglas que gobiernan la comunicación las cuales reciben el nombre de protocolo y a su vez el conjunto de protocolos forma un estándar. Los estándares fueron creados con el propósito de permitir la interoperabilidad entre dispositivos de diferentes fabricantes. Las grandes empresas actualmente han apostado por diferentes tipos de protocolos y estándares entre los más importantes tenemos: Ethernet-TCP/IP, IEC61850, y DDS.

2.2.1 Ethernet

Ethernet es definida generalmente como el estándar más popular creado para la intercomunicación de componentes de red fabricados por varias compañías, considerando la manera en que cada componente trabaja e interactúa con los demás componentes de la red. Este estándar nace con la necesidad de dar solución al problema de que dos o más hosts o equipos de red utilicen el mismo medio de comunicación y que las señales no se interfieran. A partir de 1982 los organismos de estandarización como European Carton Makers Association (ECMA), IEEE 802.3, American National Standards Institute (ANSI) y el International Organization for Standardization (ISO) fueron adoptando a Ethernet como un estándar por lo que se convirtió en la red de área local (LAN) más popular. La implementación de Ethernet debía cumplir con un gran

número de objetivos entre los cuales están: ser simple de configurar y de manejar, tener un bajo tiempo de retardo, lograr una gran compatibilidad entre los diferentes equipos, ser un sistema de comunicación estable y de fácil mantenimiento, poseer direccionamiento flexible con una arquitectura en capas y ser veloz, todo esto intentando siempre abaratar costos.

El estándar puede tener tres tipos de topologías principalmente en anillo redundante, en estrella o en bus y los datos a ser transportados por la red utilizan el método de Percepción de Portadora de Acceso Múltiple con Detección de Colisión (CSMA/CD - Carrier Sense Multiple Access with Collision Detection). La información puede viajar en una variedad de medios como: cable coaxial, fibra óptica y/o par trenzado. Además un gran número de dispositivos se ha adoptado a este estándar de comunicación como es el caso de conmutadores (switches), puentes, routers y un conjunto grande de medidores, actuadores, sensores, etc.

2.2.2 TCP/IP

TCP/IP es un protocolo comúnmente utilizado por todos los computadores conectados a Internet de manera que se puedan comunicar entre sí. Una de las ventajas más importantes de este es que se encarga de que la comunicación sea posible entre todos los elementos ya que en Internet existen ordenadores de diferentes clases, hardware y software incompatibles y una gran variedad de formas y medios de conexión. TCP/IP es compatible con cualquier sistema operativo y con cualquier tipo de hardware, además no es un único protocolo sino que en realidad es un conjunto de protocolos que cubre los diferentes niveles del modelo OSI. Los dos protocolos más importantes de este grupo son:

- Transmission Control Protocol (TCP)
- Internet Protocol (IP).

El TCP/IP funciona sobre varios tipos de red o de medios físicos proporcionados por sus propios protocolos. Por esta razón se debe considerar que los protocolos utilizados en este nivel pueden ser muy diversos y no forman parte del conjunto TCP/IP pero esto no es un problema para este protocolo. Para lograr la transmisión de información a través de este, se debe dividir el mensaje en unidades de menor tamaño llamados "datagramas" que son un conjunto de datos que se envían como mensaje independiente.

La siguiente figura muestra las siete capas del modelo de referencia OSI y su correspondencia general con las capas del conjunto de protocolos TCP/IP.

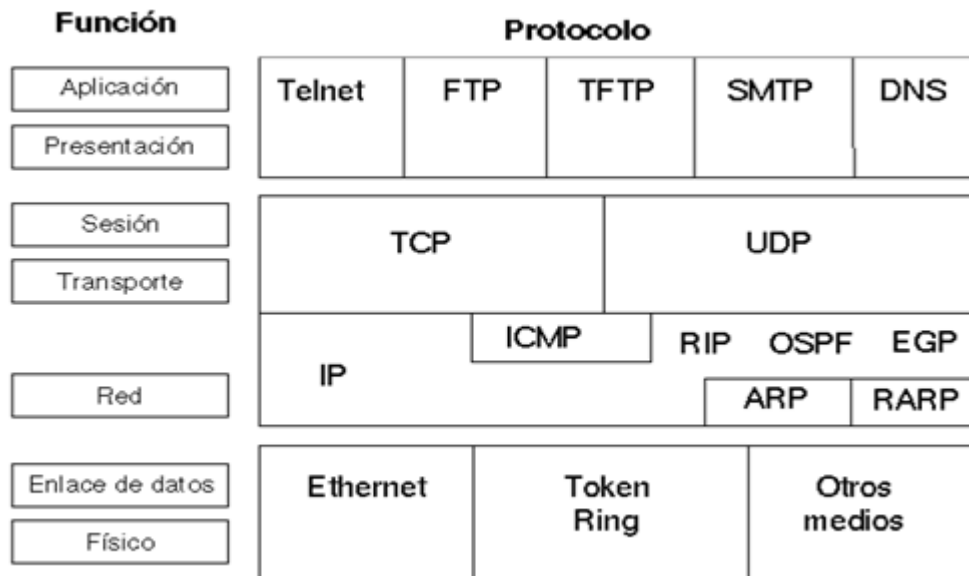


Figura 5: Modelo de referencia OSI y el conjunto de protocolos TCP/IP. Fuente: www.securenet2002.tripod.com

2.2.3 IEC61850

IEC 61850 es un sistema de automatización de subestaciones que ofrece la integración del monitoreo de estado, protección, automatización y control utilizando la información que tienen DEIs (Dispositivos Electrónicos Inteligentes) que tiene involucrado el protocolo IEC. Para cumplir con todo esto, el sistema divide la actividad en dos grupos específicos:

- El primero relacionado a la parte IEC 61850-8-1 llamado bus de estación (Station Bus) en donde se interconecta todos los dispositivos electrónicos internos inteligentes, estableciendo una red de datos entre ellos capaz de manejar de manera más racional y clasificada la información sin la necesidad de equipos adicionales (Maestros).
- El IEC también define en su modelo IEC 61850-9-2 la recolección de la información de proceso, estos datos comprenden la información que viene de la subestación a los DEIs correspondientes, a esta parte se la llama bus de proceso (Process Bus).

El estándar tiene su propia norma y los principales objetivos de esta son:

- Interoperabilidad: Intercambio de información entre los DEIs de varios fabricantes que cumplan con el estándar IEC 61850. Estos elementos usan esta información para su propia función
- Configuración Libre: Asignación libre de funciones a los aparatos. Soporta cualquier filosofía del Cliente (sistemas centralizados o descentralizados).
- Estabilidad a largo plazo: A prueba de obsolescencia. Sigue el proceso en tecnología dominante de comunicaciones. Cumple con los requerimientos de evolución del sistema necesitados por los clientes.

2.2.4 Data Distribution Service

Data Distribution Service (DDS) es la especificación del middleware² diseñado y desarrollado para estandarizar el modelo de programación de data-centric publish-subscribe para ser aplicados mayormente en sistemas distribuidos [21]. DDS simplifica la engorrosa programación de red de comunicación implementando el modelo de publicación-suscripción entre los componentes de la red para enviar y recibir datos, eventos, comandos, en resumen cualquier dato de interés. Algunos de estos componentes presentes en el sistema producen o generan información, crean tópicos (por ejemplo, temperatura, localización, presión, etc.) y publican muestras de estos tópicos, a este grupo de componentes se les denomina publicadores. Data Distribution Service entrega estas muestras a todos componentes de red que declaren su interés en el tópico en cuestión, a estos componentes se les denomina suscriptores. Cualquier integrante puede ser un publicador, suscriptor, o ambas cosas en paralelo. Con estas destrezas, DDS facilita a los desarrolladores el uso de componentes Ethernet comerciales de bajo coste para distribuir datos de forma periódica con frecuencias altas y estrictos requerimientos de tiempo de entrega. De esta forma se simplifica la integración y gestión de sistemas de tiempo real en conjunto con otros sistemas de red empresariales.

Otra importante destreza de DDS es que las aplicaciones de este sistema no necesitan información sobre el resto de aplicaciones participantes, incluyendo su existencia o localización. Algunas de las ventajas de su uso son:

- Es apropiado para sistemas de alto rendimiento gracias al protocolo RTSP (Real Time Stream Protocol) [17] que mediante una cabecera de poco tamaño ofrece gran versatilidad.
- Está basado en el modelo publicación/suscripción (publish/subscribe) lo que le otorga una gran sencillez.
- Es escalable dinámicamente.
- Tiene una arquitectura flexible y adaptable que posibilita el auto-descubrimiento de aplicaciones.
- Tiene gran número de parámetros configurables lo que proporciona a los desarrolladores un gran control del sistema.
- Brinda entregas de información con tiempos deterministas.
- Soporta comunicación unicast³ y multicast⁴ [22].

² Middleware: nombre genérico que se utiliza para el software de conectividad que está situado entre la capa de aplicación y el sistema operativo.

³ Unicast: proceso de envío de información en una o más unidades de datos (datagramas IP) desde una maquina origen a una maquina destino.

⁴ Multicast: proceso de envío de información en una o más unidades de datos (datagramas IP) desde una maquina origen a varias maquinas destino.

- Hace un uso eficiente del ancho de banda.

2.3 Sistemas de comunicación usados por empresas del mercado

En el mundo existen algunas empresas que ofrecen productos y servicios para la construcción y puesta en funcionamiento de una micro-red entre las cuales se pueden nombrar ABB, General Electric, Alstom, Spirae, Encorp, Siemens y RTI. Estas industrias del campo eléctrico (excepto RTI que es del campo de las telecomunicaciones) han desarrollado diferentes sistemas con los que se puede desarrollar y armar completamente una micro-red. La diferencia entre estos sistemas radica en el precio, prestaciones, limitaciones, tecnologías, interfaces, topologías y seguridades que pueden ofrecer. A continuación se explicará brevemente en una tabla cuáles son las principales características de comunicación de estos sistemas para la gestión de micro-redes:

- I. ABB
- II. General Electric
- III. Alstom
- IV. Spirae
- V. Siemens
- VI. RTI.

Ref.	I	II	III	IV	V	VI
	ABB	G.E.	Alstom	Spirae	Siemens	RTI
Producto	Renewable Microgrid Controller MGC 600	Grid IQ™ Microgrid Control System	Energy Management Systems ALSTOM GRID	Danish Cell Controller Project	The Siemens Spectrum Power™ TG	LocalGrid y RTI con soluciones micro-red utiliza protocolo DDS.
Características	Gestiona y controla la planta y cargas con fuentes de energías renovables y no renovables distribuidas.	Red Ethernet en puntos de medición. Para largas distancias se prefiere red inalámbrica segura o fibra óptica.	SCADA/EMS. Psymetrix y UISOL con herramientas para vigilancia de la fiabilidad de la red en tiempo real.	Arquitectura escalable, controles DERs activos, Generación Distribuida (DG) y energías renovables.	Sistema flexible fiable, expansible con control de la red de alimentación. Incluye todas las funciones Sistema de Gestión (EMS).	LocalGrid utiliza DDS para el transporte entre nodos distribuidos, para asegurar y optimizar los elementos.
Conectividad	Sin información	Dispositivos con Modbus TCP/IP que interactuar con las unidades de generación o de almacenamiento.	Posee PMU (Unidad de medida fasorial), manejo de Terminales Remotos (RTUs).	Sistema de seguimiento celular (CMS), con gestión y control de la infraestructura, tiempo real.	Disponible en Windows ® y Plataformas de sistemas operativos Linux ®.	Comunicación desde la LAN (red de área local) a la banda ancha y comunicaciones satelitales.
Protocolo y Estándar	Ethernet, Modbus, CAN-Bus y otros. Cableado de E/S (digital, analógico) para conectar dispositivos.	Modbus TCP/IP como maestro. Controlador D400 como puerta de enlace. Protocolo de comunicaciones DNP3.0.	Incluyen; TCP/IP, ActiveX, ODBC, DDE y otros. Protocolos de intercambio de datos como: ICCP, DNP 3.0, TCP/IP & Serial, Modbus.	Posee Internet TCP/IP and SMS. Protocolo BCD. Señales analógicas y digitales por cable.	Incluye IEC 61850, dial-up, conmutación de paquetes dedicada, de fibra, de radio, y TCP/IP y medios de comunicación WAN.	LocalGrid traducen DNP3, IEC 61850 y comunicaciones SCADA en DDS a través de una topología distribuida.
Seguridad	Sin información	Contraseñas para: acceso local, acceso remoto y control de acceso remoto.	Autenticación basada en Kerberos y funciones de autorización y auditoría.	Incluye seguridad cibernética NIST propuesta, adaptada de los EE.UU.	Posee NERC (Norma de seguridad) y normas de acción (CIP 002-009).	Conectividad VPN. Fundamento Linux. Todos los puertos no críticos se eliminan.
Arquitectura	Sin información	Sin información	Arquitectura flexible y escalable.	Arquitectura de control distribuida.	Sin información	Arquitectura distribuida.

Tabla 2: Empresas que ofrecen la implementación y comunicación de la micro-red.

En la tabla 2 se puede ver que cada empresa presenta su tipo de solución orientada a micro-redes con diferentes características, componentes y métodos de comunicación, por lo que se puede decir que hoy en día en el mercado existen varios tipos de sistemas para el monitoreo y control pudiendo ser conectadas a fuentes de energías renovables como generadores eólicos y generadores fotovoltaicos. Otra de las características importantes de estos sistemas es la posibilidad de trabajar en tiempo real, es decir lograr grandes velocidades con la implementación de interfaces de comunicación que soportan la distribución de la información por toda la red.

Las empresas han visto la posibilidad de tener elementos que interconecten y realicen medidas en cada estación de generación o almacenamiento de energía, tales dispositivos pueden ser: Unidades de Medida Fasorial (PMUs) en (III), Unidades de Terminal Remoto (RTUs) en (III, V), Recursos de Energía Distribuidos (DERs) en (IV) y otros dispositivos inteligentes que hacen que la comunicación sea fluida, logrando que la información esté disponible cuando y como se la requiera.

Los bits de información que están circulando por toda la red de datos mantienen cierto estándar y protocolo para que el intercambio de datos se realice de manera correcta, estos protocolos son diferentes de una empresa a otra pero entre los más destacados e importantes están Ethernet TCP/IP (I, II, III, IV, V, VI) el cual es un protocolos que se está utilizando en la actualidad cada vez más en sistemas para micro-redes por su velocidad y simplicidad, Modbus (I, II, III,) que es uno de los protocolos más antiguos aun usados por las empresas por su simplicidad de manejo pero que ha perdido mercado por su baja velocidad y alto procesamiento. La empresa RTI implementa hace algunos años DDS sobre Ethernet (VI) para soluciones de comunicación de diferentes proyectos grandes y promueve la velocidad en tiempo real, aplicaciones IoT y muchas otras ventajas mas que se recalcaran más adelante.

2.4 Mejores sistemas de comunicación en micro-redes

Con el propósito de comparar los diferentes sistemas de comunicación para micro-redes se ha descrito anteriormente los estándares más usados por las industrias en forma resumida. Teniendo un breve conocimiento de estos es posible analizar cada uno, observar sus ventajas y desventajas para finalmente llegar a un consenso y decidir de forma eficaz el mejor sistema de comunicación para micro-redes.

Entre los sistemas comúnmente utilizados esta Modbus y sus variantes, este no es un estándar como tal sino un protocolo de mensajes utilizado desde algunos años atrás en sistemas de comunicación específicamente para la automatización y control de equipos. Funciona siempre en modo maestro-esclavo lo cual constituye la principal desventaja de este sistema debido a que el maestro debe preguntar para que determinado esclavo responda, caso contrario no existe el intercambio de datos. De esta forma se está haciendo un mal uso de la red disminuyendo el rendimiento de la red y el tiempo de respuesta de los datos.

El siguiente protocolo es DNP3 que al igual que Modbus trabaja en modo maestro-esclavo y poseen topologías tipo bus solamente. Este ofrece flexibilidad, robustez y comunicación desde

estaciones remotas, adicionalmente permite la asignación de prioridades a los mensajes por todo esto se podría decir que DNP3 es una evolución o mejora de Modbus. Cabe destacar que estos dos protocolos son no propietarios por lo que se puede implementar libremente.

En la lista a continuación aparece Ethernet que trabaja a nivel físico y de enlace de datos en el modelo OSI. Existen variantes del mismo estándar que pueden tener diferentes velocidades de transmisión desde 10Mb/s por cable coaxial pasando por transmisión por par trenzado a velocidades de 100 o 1000Mb/s hasta llegar al intercambio de datos por fibra óptica alcanzando velocidades hasta de 10Gbit/s. Con todas estas características se puede afirmar que Ethernet tiene un sistema de comunicación mucho más elaborado y con mejores características y prestaciones que Modbus o DNP3.

El siguiente en la lista es TCP/IP el cual es un conjunto de protocolos que cubre los diferentes niveles del modelo OSI, además trabaja a nivel de red y transporte dentro del modelo OSI y puede ser transmitido por Ethernet. Al igual que Ethernet, TCP/IP ofrece muchas ventajas en el intercambio de información como se menciono anteriormente es por esto que la unión de Ethernet y TCP/IP sería una buena solución para las comunicaciones actuales de micro-red.

El próximo en la lista para sistemas de comunicación aparece la norma para la gestión y control de subestaciones IEC 61850 el cual trabaja sobre Ethernet alcanzando capacidades desde 100Mbps hasta 1Gbps. Además se caracteriza por tener un modelaje de los datos que permite de una forma más estandarizada entender (independiente del fabricante) cual es la información que vamos a transmitir. La mayor ventaja de IEC es la estandarización y configuración libre lo que permite trabajar con dispositivos certificados con la norma en micro-redes con el propósito de realizar la configuración de los equipos de una manera más rápida y automática

Por último pero no menos importante esta la especificación DDS el cuales está basado en el modelo de comunicación publicador-suscriptor presente en la red y sus componentes, además es usado mayormente para aplicaciones distribuidas. Los publicadores ponen a disposición los datos en la red para que los suscriptores interesados recojan esta información y la procesen. Brinda la posibilidad de comunicación en tiempo real lo que hace de DDS una herramienta eficaz para ser usado en micro-redes. Cabe recalcar que esta especificación puede trabajar con la norma IEC61850. Adicionalmente DDS trabaja sobre la red y con componentes compatibles con Ethernet de esta forma se simplifica la integración y gestión de sistemas y componentes dentro de las micro-redes eléctricas.

	Modbus	DNP3.0	Ethernet	TCP/IP	IEC61850	DDS
Modo o modelo de Comunicación	Maestro / Esclavo	Maestro / Esclavo	Cliente / Servidor	Cliente / Servidor	Cliente / Servidor - Publicador / Suscriptor	Publicador / Suscriptor
Datos en tiempo real	NO	NO	SI	NO	SI	SI
Calidad de Servicio	NO	NO	SI	SI	SI	SI
Max. Velocidad de datos	Aprox. 2Mbps	Aprox. 2Mbps	10Gbit/s	10gGbit/s	1Gbps	10Gbit/s
Medio de Comunicación	Par trenzado, fibra óptica, radio	Par trenzado, fibra óptica, radio	Coaxial, Par trenzado, fibra óptica	Coaxial, Par trenzado, fibra óptica	Coaxial, Par trenzado, fibra óptica	Coaxial, Par trenzado, fibra óptica
Para aplicaciones IoT	NO	NO	SI	SI	NO	SI

Tabla 3: Características de los sistemas de comunicación para micro-redes.

Los datos comparativos de la tabla 3 muestran los resultados de los sistemas para la comunicación entre los elementos de la micro-red. Cabe recalcar que la norma IEC61850 puede ser incluida en el estándar de comunicación DDS lo que hace de esta unión la mejor solución para la implementación de la comunicación en las micro-redes. Los siguientes dos capítulos tratan de enfocar el estudio del sistemas de comunicación con el estándar DDS, de esta manera se podrá analizar más a fondo el diseño, principio de implementación y la puesta a punto de este sistema anteriormente mencionado.

3 CAPITULO 3 ESTANDAR DE COMUNICACION DDS PARA MICRO-RED

Este capítulo recopila las bases de la teoría del estándar DDS y menciona como el estándar puede acatar las necesidades actuales que tienen las micro-redes. Así mismo se detallarán algunos aspectos relevantes sobre como DDS puede ser implementado en sistemas SCADA y diferentes soluciones en redes para finalizar recalcando los proyectos industriales llevados a cabo con el estándar.

3.1 Bases de la teoría del estándar DDS

DDS (Data Distribution Service) es un conjunto de estándares de la OMG (organización sin fines de lucro que promueve el uso de tecnologías orientadas a objetos mediante guías y especificaciones) que promueve y gestiona una comunicación en tiempo real basada en publicación y suscripción [19]. Surge de sistemas navales de altos requerimientos y lo que busca principalmente es la comunicación instantánea. Una de las características de este estándar es que existe una difusión de datos, es decir los datos residen en toda la red y no en elementos privilegiados, esto no quiere decir que todos los elementos de la red tengan todos los datos posibles. Cada elemento de la red tiene información de aquellos datos que va a publicar y se suscribe a aquellos datos de su interés, por lo que cada elemento no tiene que recibir la información que no le pertenece, es decir la red puede tener varios dominios disjuntos lo cual es muy importante para seguridad [18].

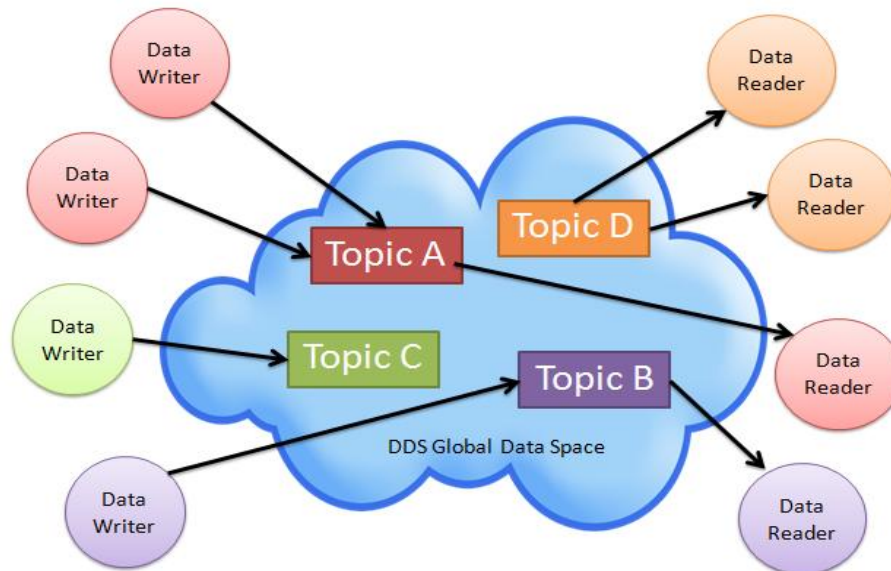


Figura 6: Esquema básico de comunicación DDS Fuente: www.primstech.com.

En la figura 6 se puede observar el espacio global de datos DDS representado por la nube, en la cual existen tópicos los mismos que serán publicados o escritos en este espacio por los escritores de datos o publicadores. De la misma manera existirán lectores o suscriptores que podrán acceder

a esta información tan solo suscribiéndose al tópico en cuestión para recibir los respectivos datos. Cabe recalcar que cualquier nodo puede ser publicador o suscriptor o ambos a la vez.

Los estándares de la OMG para DDS son dos:

- DDS v1.4. Especificación DDS que describe el modelo de centro de datos de publicador y suscriptor (DCPS) para la comunicación de aplicaciones distribuidas y la integración [19].
- DDSI/RTPS v2.2 Define el protocolo de tiempo real entre publicadores y suscriptores (RTPS) interoperabilidad del protocolo DDS [20].

3.1.1 Metodología de comunicación DDS

Toda la comunicación de DDS esta desacoplada en datos, existe una red de datos común pero cada participante saber lo que va a enviar o recibir. Los datos están agrupados en tópicos de información, no se envían datos al azar sino que se envían datos agrupados con un sentido lógico. Cada elemento de la red da a conocer lo que quiere enviar y pide lo quiere recibir pero no sabe quién envía los datos o cuando. Se puede dar el caso que un elemento o sensor envíe información que nadie quiera suscribirse a este en ese instante, por ejemplo un sensor de humedad puede estar enviando información y nadie esté presente ese momento para recibir este dato, o puede darse el caso de que un elemento este esperando la recepción de información de estado de apagado o prendido y nadie le está transmitiendo esa información.

Los publicadores y suscriptores son ajustados automáticamente y dinámicamente por el descubrimiento dinámico de DDS, por lo que se permite que cualquier elemento entre en cualquier momento dentro de la red y que tenga los mismos derechos que los demás. En pruebas realizadas en redes muy lentas, con cortes y con poco ancho de banda, el tiempo de descubrimiento ha sido aproximadamente de 4 segundos en algunos casos y antes de este tiempo ya se podía enviar y recibir la información sin ninguna clase de problemas [18]. El propio DDS se ocupa de los detalles de saber quién está enviando, quién está recibiendo, con qué calidad, etc.

Cada elemento de este espacio conoce lo que le interesa saber, así mismo todo es dinámico por lo que no hace falta empezar declarando la publicación. El espacio global consiste en una base de datos distribuida en la que los publicadores insertan nuevos datos o actualizan datos existentes y los suscriptores informan con filtros pertinentes estas inserciones y actualizaciones. Cada componente o nodo de la red es un elemento de la base de datos flotante de memoria y DDS se encarga de sincronizar esta información distribuida a través de lo que se publica y se suscribe. También existe la posibilidad de sincronizar esta base de datos flotante con una base de datos física.

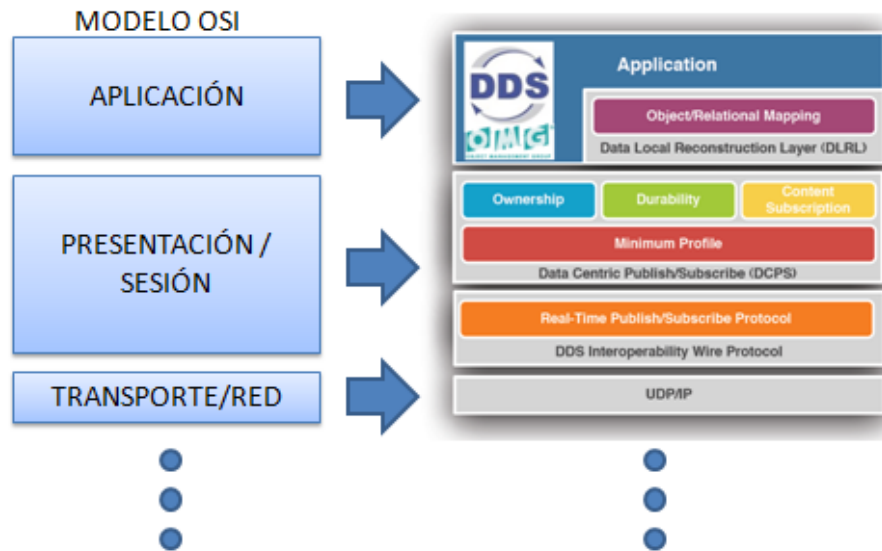


Figura 7: Comparación de DDS con el Modelo OSI. Fuente: Propia

En la figura 7 se indica la presencia de DDS en el modelo OSI y se observa que este estándar está presente en las capas de aplicación y presentación. De manera simplificada las capas de DDS las cuales son DLRL y DCPS con sus funciones respectivas también se indican. La capa DDS Interoperability Wire Protocol pertenece a la parte de RTPS (Real Time Publisher Subscriber) que es el protocolo del estándar. En la parte inferior de la grafica se puede observar las capas de transporte y red las cuales trabajan con UDP e IP respectivamente. Las capas del estándar se explicaran con más detalle en el siguiente punto.

3.1.2 Capas DDS

DDS se divide en dos capas el Data-Centric Publish-Subscribe (DCPS) que es el responsable de la distribución de los datos y el Data Local Reconstruction Layer (DLRL) que es la capa responsable de adaptar los datos a las aplicaciones locales. En un sistema DDS la capa DCPS es obligatoria, mientras que la capa DLRL es opcional dependiendo de la necesidad de adaptación de la información que tengan los componentes que se encuentren por encima del middleware.

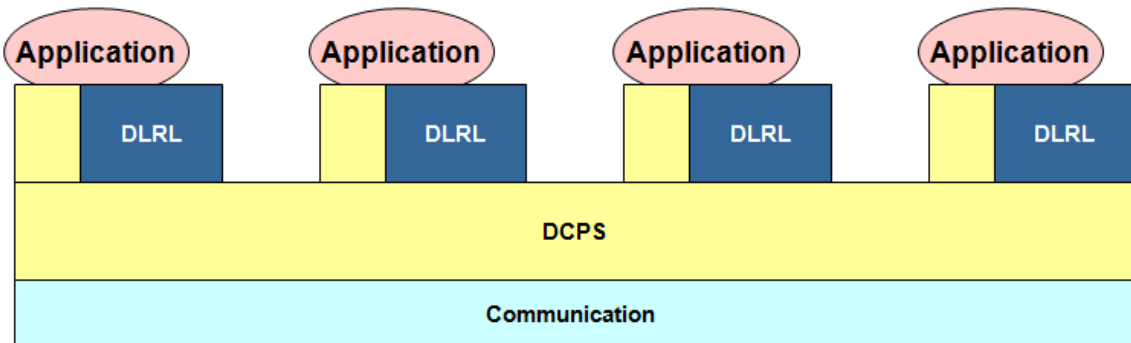


Figura 8: Ubicación de cada capa del sistema de comunicación DDS. Fuente: www.cs.wustl.edu/~schmidt/cs395/DDS-intro.ppt

En la figura 8 se ven las dos capas que forman parte del sistema DDS, además muestra la utilidad de cada capa en el sistema DDS. La capa DLRL está estrechamente relacionada con las aplicaciones que vayan a ser comunicadas con esta tecnología, en cambio la capa DCPS tiene en su interior características para una comunicación efectiva entre las cuales destaca la durabilidad, la suscripción, la titularidad, provee un espacio global accesible a todas las aplicaciones de los interesados, estable calidades de servicio y proporciona descubrimiento y configuración automática. A continuación se detallara un poco más cada una de estas capas.

DCPS

La capa DCPS es responsable de difundir de manera eficiente los datos de los publicadores a los suscriptores interesados. Se implementa utilizando los conceptos de publicador o escritor de datos en el lado emisor y suscriptor o lector de datos en el lado receptor. La capa DCPS se compone de uno o más dominios de datos, cada uno de los cuales contiene un conjunto de participantes (publicadores y suscriptores) que se comunican a través de DDS. Cada entidad (es decir, publicador o suscriptor) pertenece a un dominio [21]. Cada proceso tiene un dominio participante (Domain Participant) para cada dominio de datos del cual es miembro.

Dentro de la capa DCPS existen también elementos bien diferenciados que hacen posible el buen funcionamiento de esta capa y de todo el sistema en conjunto. A continuación se describe brevemente cada uno de estos.

- Participante de dominio (DomainParticipant): representa la participación de una aplicación dentro del conjunto de comunicaciones del sistema.
- Tópico (Topic): Describe los datos asociados a un tópico y se usa para asociar los lectores a los escritores. Se identifican con una cadena de texto a la que luego se le enlaza un tipo de dato.
- Escritor (Data Writer): Aplicación que permite el acceso para escribir datos tipados sobre un tópico particular.
- Publicador (Publisher): Es la estructura responsable de un conjunto de escritores. Su tarea es repartir la información.
- Lector (Data Reader): Aplicación que permite el acceso para leer datos tipados relativos a un tópico específico.
- Suscriptor (Subscriber): Es la estructura responsable de un conjunto de lectores responsables de recibir información [22].

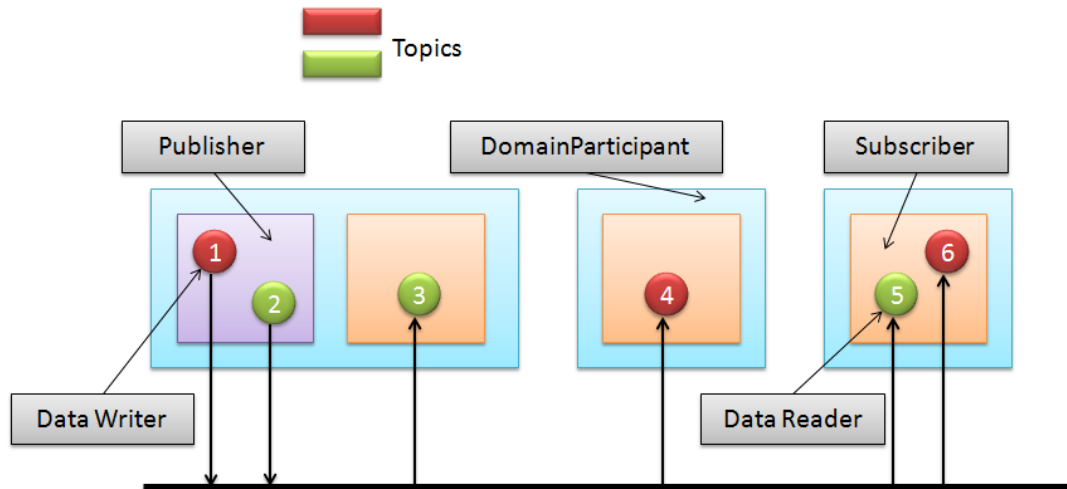


Figura 9: Elementos de la capa DCPS. Fuente: www.primstech.com.

En la figura 9 se puede ver la ubicación de cada uno de los elementos que conforman la capa DCPS de DDS, así como su función dentro del intercambio de datos en el sistema y la dirección de flujo de datos o de tópicos a la red de comunicación. Ejemplos de tópicos pudieran ser la temperatura de una caldera, la velocidad del viento, el voltaje de un generador y muchos otros tipos de datos.

Como resumen, cuando una aplicación desea publicar una información, lo hace escribiendo en un “Topic”, a través de un objeto “Publisher”. Este objeto reside en un “DomainParticipant” que gestiona tanto los objetos “Publisher” como los “Subscriber”, que son los responsables de recibir los mensajes. Los objetos “Publisher” proporcionan los mensajes a petición de la aplicación, mientras que los objetos “Subscriber” avisan a la aplicación de la llegada de un mensaje. En este último caso la iniciativa de la comunicación la toma el sistema de comunicaciones y no la aplicación. De esta manera para que la comunicación sea efectiva se debe tener entre un publicador y un suscriptor un mismo dominio, un mismo tópico y un mismo tipo de dato, así en la figura será posible tan solo la comunicación efectiva de 2 a 3.

DLRL

Es la capa opcional que se puede colocar por encima de la capa DCPS. Su propósito es proveer de un acceso más directo a los datos intercambiados y aportar los beneficios de la orientación a objetos así como permitir una fácil integración de los tipos de DDS con los tipos que necesite nuestra aplicación. Mediante la utilización de DLRL el desarrollador conseguirá:

- Describir clases de objetos con sus métodos, campos y relaciones.
- Enlazar los campos con las entidades DCPS
- Manipular los objetos (crear, borrar, leer/escribir) usando el lenguaje nativo.
- Mantener esos objetos en una cache, asegurándose de que las referencias que apuntan a esos objetos no se pierdan en ningún momento.

3.1.3 Calidades de servicio en DDS.

La calidad de servicio de DDS es otra característica muy importante del sistema ya que comprenden las cualidades de la información y las calidades de la distribución, como por ejemplo si se está enviando la presión de un sensor, se tendría que considerar en qué clase de red se está trabajando para darles la característica de calidad a los datos a enviar, es decir se puede decidir cómo se quiere enviar cierta información detalladamente. Además DDS posee 22 calidades de servicio (QoS) para controlar las circunstancias de la comunicación, de esta manera se tiene la información y la forma en la que se controla como gestionar esta información [18]. La calidad de servicio va por cada tópico que se envía en cada elemento que lo envía, no es general sino que se puede dar una calidad a cada elemento en particular y existen 4 categorías en general las cuales son:

- **Control de los recursos locales:** si un nodo físicamente está en una máquina que es peor, que tiene menos memoria por ejemplo, se puede establecer una calidad de servicio que determine cuánto recurso puede aguantar, con esto es posible acordar que si la memoria llega a un 80% deje de admitir mas información. Si un nodo se cae no afecta mayormente a la red ya que como todo es distribuido otro nodo puede tomar su lugar inmediatamente debido a la existencia de tolerancia a fallas.
- **Control espacial de la distribución de datos:** Existen tipos de información que admite la pérdida de cierto número pequeño de datos y otra que no, esto también es posible controlar y por tanto se puede decidir la forma en la que viajan los datos y su orden, por ejemplo la transmisión de video en tiempo real está compuesto por datos que admiten la pérdida de algunos bits sin problema.
- **Ciclo de vida de la información:** se puede enviar un tipo de información que permanezca en la red, de tal manera que cualquier otro nodo que llegue después sea capaz de obtener esta información, así como también se puede almacenar de manera histórica la información en la red.
- **Otras características relacionadas con el tiempo de la información:** por ejemplo es posible exigir que alguna información llegue antes de cierta cantidad de tiempo o podemos pedir que si una información llega después de cierto tiempo se considere no válida.

Muchas calidades de servicio además se pueden ajustar según un modelo de oferta y demanda. El que envía la señal declara la calidad con la que envía y el que desea recibir declara con que calidad la quiere recibir y si la información no tiene la calidad adecuada no se recibe por el criterio de oferta y demanda. Esto permite un filtrado de datos.

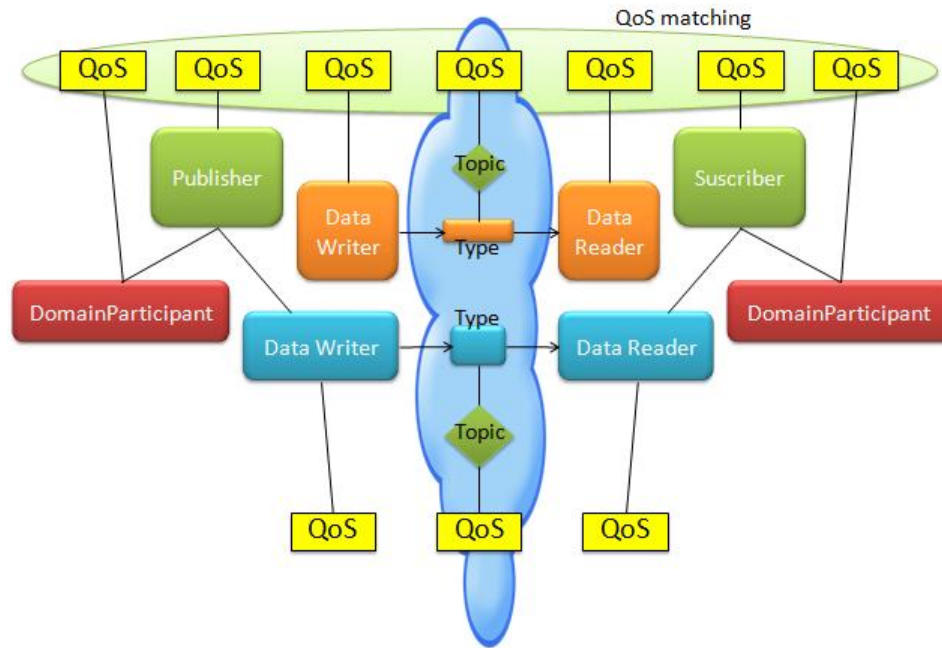


Figura 10: Calidades de Servicio presentes en componentes DDS. Fuente: www.primstech.com.

La figura 10 muestra las respectivas calidades de servicio (QoS) que tienen cada una de las partes del sistema como son los dominios independientes, los publicadores, los suscriptores y en cada uno de los tópicos también se hace presente las calidades de servicio como se indicó anteriormente. La figura muestra además la conformación del sistema referente a la parte de calidad de servicio que ofrece DDS para gestionar de mejor manera la información, tener respaldo de los datos, hacer un uso eficiente de la tolerancia a fallas, entre otras características de este sistema.

3.2 DDS para solucionar retos SCADA.

3.2.1 Soluciones en Datos

Cuando se analizan los datos en DDS, estos están orientados a trabajar en tiempo real, es por esta razón que es conveniente para sistemas SCADA (Supervisión, Control y Adquisición de datos).

Además de ser rápido en la transferencia de datos posee ciertas características que destacan en este sistema como por ejemplo el filtrado de datos de dos tipos principalmente:

- ✓ Por contenido
- ✓ Por frecuencia de actualización de cada parámetro.

Se puede así mismo filtrar datos por características de calidad de servicio y recepción de datos haciendo que los datos se puedan controlar y administrar de mejor manera sea en el envío como en la recepción. Por ejemplo si un sensor está enviando datos cada 10ms se puede configurar el sistema para que solo reciba este dato de este sensor en concreto cada 1s, así el receptor posibilita filtrar los datos para analizarlos eficientemente. DDS adicionalmente incorpora el

concepto de difusión de datos lo que permite sumar o agregar nuevos datos de información en cualquier momento, así la información puede entrar o salir de la red en un instante dado.

Otra de las características de DDS es la sincronización de información desde distintas fuentes con esto por ejemplo se puede sincronizar diferentes datos de video de distintas cámaras y recibirlas en una sola o se puede recibir datos de distintos sensores de temperatura y en pantalla mostrar tan solo una temperatura promedio o global de esa medida. Así este sistema permite que los datos estén distribuidos en tiempo y en espacio, conoce qué se recibe, cómo se recibe y no necesita saber que dispositivo envía la información.

3.2.2 Soluciones en Redes

Las calidades de servicio permiten evitar los problemas en redes lentas o de malas condiciones. Es posible parametrizar con calidades de servicio casi todo en cada tipo de sistema, desde el tamaño de los paquetes pasando por el número de veces que se reenvía un paquete en caso de pérdida, hasta el tamaño de las banderas que se utilizan para saber si un nodo remoto sigue vivo. Con diferentes pruebas que se han elaborado con este sistema, se ha podido comprobar que el envío de datos en redes lentas con poco ancho de banda y múltiples cortes es posible transferir información incluso de video como por ejemplo video sobre Wi-Fi. En el tema de redes DDS posee flexibilidad y escalabilidad, desde un sistema embebido⁵ hasta redes de redes. Además de cubrir redes lentas, el sistema está evolucionando de tal manera que se está implementando ciertos mecanismos como por ejemplo el envío de un mensaje de respuesta de confirmación de dato recibido no por cada dato recibido sino por un grupo de datos, así también se puede realizar Quorum Writers es decir, si la información de 5 sensores se pretende enviar conjuntamente, no se enviara ésta si los datos de todos los sensores no está disponible [18].

En redes mas grandes ha habido bastante evolución como es el caso de la implementación del descubrimiento de elementos, de tal manera que si una red se conecta con otra red se puede descubrir los dispositivos disponibles en la otra red. Este proceso es realizado automáticamente por DDS siendo para el usuario muy conveniente a la hora de la configuración de cada elemento de red y sus diferentes parámetros o medidas convirtiéndose hoy en día en unos de los estándares más influyentes para el Internet de las cosas (IoT).

3.2.3 Solución en Integridad de datos

La integridad de datos se establece en la propia comunicación, si se declara un dato como fiable o se declara un Quorum Writer y se establece para ser enviado, el otro extremo recibirá esta información, DDS se encarga de ver cómo el sistema realizará el intercambio de estos datos y qué método utilizará para esto, es decir algunas veces el sistema tendrá que reintentar o tal vez volver a mandar toda la información, enviar por otro método u otro mecanismo para que de alguna manera lleguen tales datos a su destino correctamente. Además existe la posibilidad de precisar

⁵ Embebido: Sistema diseñado para realizar una o algunas funciones dedicadas, frecuentemente un sistema en tiempo real.

una calidad de servicio pero hay que recordar que la integridad la mantiene la propia red de datos DDS.

Acompañado a esto como parte del estándar está la seguridad, con esta existe la posibilidad de hacer un cifrado de alta calidad como por ejemplo algoritmos de 128bits, etc. También está disponible la autenticación y separación física en dominios de tal forma que la información puede ser separada en elementos disjuntos, de esta manera ciertos nodos no podrán ver nada de la información que no les interesa logrando una mejor organización y aprovechamiento de los datos en la red [18]. DDS está avanzando hacia nuevos temas de seguridad cada vez más modernos. Adicionalmente hay que tomar en cuenta que si se realiza por ejemplo un cifrado, este compromete el tiempo y el rendimiento de la red jugando en contra a los datos de tiempo real, así como también al configurar un cifrado muy riguroso en un servidor se está creando un cuello de botella para el flujo de datos malogrando la red y la distribución.

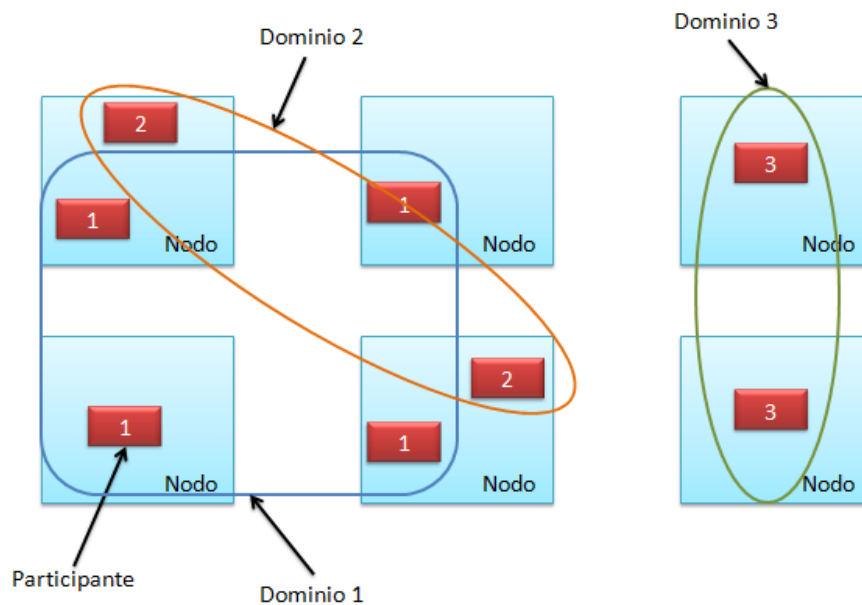


Figura 11: Participantes, nodos y dominios de DDS. Fuente: www.primtech.com.

La figura 11 muestra algunas de las partes que constituyen una red de datos DDS en la que claramente se pueden apreciar los participantes, los cuales pertenecen a diferentes nodos que no son más que la unión o intersección de un conjunto de participantes de un mismo lugar o zona. Además se puede observar como el sistema puede agrupar a diferentes participantes y formar lo que se llama un dominio.

Otra característica que sobresale en este sistema es la tolerancia a fallas, ya que DDS las proporciona en múltiples niveles. En cada participante o en cada nodo en concreto existe una calidad de servicio con respecto a tolerancia a fallas, de tal manera que si el nodo cae se puede hacer que se vuelva a levantar o que se vuelva habilitar automáticamente. En la propia red también está presente la tolerancia a fallas y como está presente la difusión de datos no existirán puntos críticos. Adicionalmente se puede hacer Plug and Play, todo se puede añadir dinámicamente por tanto se puede realizar una tolerancia a fallos involucrado con este punto.

Cada uno de los participantes en DDS de alguna manera es consciente de quienes están enviando la información que desean, por esta misma razón es que los participantes envían banderas pequeñas con el propósito de averiguar si está vivo el nodo del otro extremo ya que si no responde en cierto tiempo los declara muertos o no presentes.

3.2.4 DDS en la Industria

DDS se utiliza en muchos ámbitos industriales y surgió en principio como sistemas navales de tiempo real. Con el paso de los años este sistema ha ido evolucionando muy rápidamente en muchos ámbitos industriales. Hoy en día DDS es usado entre otras cosas para:

- Modelación y Simulación
- Sistemas Embebidos
- Sistemas SCADA de todo tipo
- Smart Grids
- Agentes Inteligentes
- Internet de las Cosas
- Formula 1
- Comunicación Machine-to-Machine (M2M).
- Cuidados de la Salud y Dispositivos Médicos
- Transporte
- Defensa y Espacio Aéreo
- Servicios Financieros
- Sistemas No Tripulados

Se puede transmitir por DDS imágenes, video, voz, datos de aplicaciones, señales industriales, y muchos otros tipos de comunicaciones todos estos en real. Se puede también desarrollar proyectos I + D y aplicaciones de uso militares ya que DDS tiene muchas ventajas. Ejemplos en concreto del uso de este sistema son: la Presa Grand Coulee en USA que es la quinta central hidroeléctrica más grande del mundo, Chemtech/Siemens en Brasil que es el diseño e implementación de un sistema de supervisión y adquisición de datos distribuidos en oleoductos y gasoductos, The Power Grid Control en USA y el Sistema de comunicación del Metro de Amsterdam por citar algunos. Las aplicaciones en la industria han despertado el interés en empresas u organizaciones que ofrecer este tipo de comunicación con DDS para solucionar diferentes sistemas industriales. Entre las empresas más importantes que ofrecen sistemas de comunicación DDS están: OpenDDS, Vortex y RTI.

3.2.5 DDS como estándar de comunicación en micro-redes.

DDS es un estándar de comunicación muy utilizados desde muchos años atrás hasta la actualidad, es decir tiene una amplia trayectoria y ha logrado ser implementado para resolver muchos sistemas de comunicaciones exigentes, demostrando así una gran evolución. Con el paso de los años como se mencionó en un apartado anterior, DDS ha incursionado en algunas tecnologías modernas en los campos del transporte, medicina, Internet de las cosas y sin duda también ha

estado presente en las micro-redes y Smart Grids. DDS ha brindado soluciones de comunicación e innovación en micro-redes, tecnología que es muy útil en estos tiempos para la sociedad y el mundo debido a que ayuda a dotar de energía eléctrica a muchas zonas aisladas y a mejorar la eficiencia energética en lugares que ya cuentan con este servicio. Hoy en día este tema está de moda por lo que en muchos países se escucha hablar de la conciencia humana acerca del ahorro energético, la eficiencia energética y fuentes de energía renovable, una micro-red comprende todos estos puntos y más.

En la actualidad existen dos empresas que ofrecen un sistema de comunicación para una micro-red con el estándar DDS las cuales son Vortex y RTI. Estas empresas han desarrollado su propio software para la gestión y manejo del estándar en el cual es posible realizar pruebas de envío y recepción de datos, así como mediciones del flujo de datos. RTI ofrece una solución para micro-redes con su sistema LocalGrid Technologies.

4 CAPITULO 4 PRUEBA DE CONCEPTO DE MICRO-RED CON DDS

En este capítulo se describe en detalle cómo se realizó una pequeña red de laboratorio para llevar a cabo las pruebas de concepto del estándar DDS. También se explicará la función de los componentes que integran la red, las distintas instalaciones, configuraciones y conexiones para la transmisión de la frecuencia eléctrica por la red de comunicación. Por último se verán algunas consideraciones importantes para realizar las primeras pruebas de aplicación.

4.1 DDS + Raspberry Pi

Luego de definir en qué consiste un sistema de comunicación DDS, observar y analizar la metodología de comunicación y las partes que constituyen este estándar, se procederá a la instalación de DDS en computadoras con sistema operativo Linux y en dos Single Boards Computers (SBCs) que trabajan con un sistema operativo Raspbian. Para realizar esto es importante conocer primeramente que es un SBC llamado también Raspberry Pi o computador de una placa, analizar cuáles son sus funciones, características y partes que lo constituyen.

Un Raspberry Pi es un ordenador o computadora muy económico y del tamaño de una tarjeta de crédito que puede conectar a un monitor o televisión y utilizar un ratón y teclado estándar. Este dispositivo permite a las personas explorar el mundo de la computación, aprender a programar en algunos lenguajes como C++ o Python. Es capaz de hacer todo lo que espera que un ordenador de sobremesa lo haga, desde navegar por Internet y reproducir vídeo de alta definición hasta trabajar con hojas de cálculo, procesador de textos, y jugar juegos. A mas de todos estos cualidades la Raspberry Pi tiene la capacidad de interactuar con el mundo externo, ya sea por sus puertos de comunicación estándar, sus entradas y salidas tanto digitales como analógicas, puertos de cámara digital, salida de sonido, USB, etc.



Figura 12: Raspberry Pi o Single Board Computer (SBC). Fuente: www.raspberrypi.org

En la figura 12 se observa el SBC, así como también algunas de sus partes o componentes que la constituyen. Este modelo se denomina B pero en la actualidad existen algunas versiones más modernas como la Raspberry Pi B+, Banana Pi y hace poco se lanzo la versión 2 de este SBC el cual tiene prestaciones mejoradas. Tiene una gran ventaja al ser un computador completo que posee el puerto Ethernet para la transmisión de datos los mismos que pueden ser leídos desde las

entradas tanto analógicas como digitales que posee. Para que este mini computador realice cada una de sus función correctamente es necesario la instalación de un sistema operativo, adicional a esto se instalará la versión DDS compatible para luego realizar las pruebas de comunicación respectivas.

Cada uno de los pasos para la instalación del sistema operativo y el estándar de comunicación en el SBC se detallan más adelante en los siguientes puntos. La versión compatible de DDS para la Raspberry Pi se puede encontrar en dos de las empresas que ofrecen este producto y son Vortex y RTI. En este caso se dispondrá de la versión DDS de RTI denominada Connext DDS Professional para descargarla del sitio web www.rti.com [23] y proceder con su instalación.

4.1.1 Instalación de DDS en el computador

Ver anexo A

4.1.2 Instalación de DDS en Raspberry Pi

Ver anexo B

4.1.3 Código C/C++ del cálculo y transmisión de la frecuencia en Raspberry Pi

Ver anexo C

4.2 Red DDS para pruebas de comunicación

Una vez terminados los procesos de instalación y configuración del estándar de comunicación DDS en los computadores y en la Raspberry Pi, se procederá a la realización de una red de comunicación tipo estrella para probar la transmisión y recepción de datos. Para lograr esto se dispondrá de 3 computadores, dos Raspberry Pi y un switch o router. Los datos que pasarán por la red serán datos de frecuencia de la red eléctrica (50Hz) para de cierta forma simular la medida de tal parámetro en las micro-redes. El SBC por medio de un circuito electrónico será capaz de adquirir la señal analógica de la frecuencia para luego digitalizarla, calcularla y enviarla a la red de comunicación del estándar DDS.

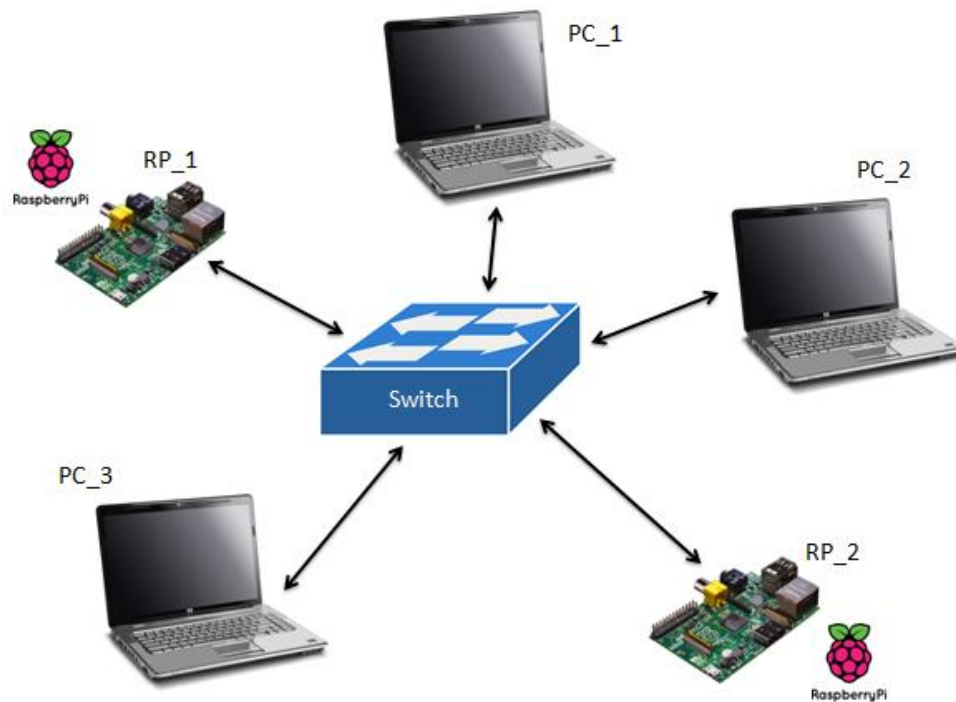


Figura 13: Topología de la red DDS para pruebas y mediciones. Fuente: Propia.

La figura 13 muestra la topología de red que se utilizará para la implementación del estándar de comunicación DDS que servirá a su vez para realizar las respectivas pruebas de envío y recepción de datos entre los distintos componentes de red. En las pruebas a realizar los SBCs serán los que envíen los datos a las computadoras, de esta forma se podrá observar el intercambio de datos a través de este estándar. Para las pruebas que se realizarán a continuación con la transmisión de la frecuencia de la red eléctrica, las Raspberry Pi serán los publicadores y los computadores serán los suscriptores. Cabe recordar que un mismo equipo puede ser publicador, suscriptor o los dos a la vez.

En la red DDS cada componente dispondrá de una IP perteneciente a la red en cuestión, además en esta misma red es posible conectar mas dispositivos publicadores o suscriptores, tan solo hace falta instalar y configurar DDS en estos nuevos equipos para que puedan integrarse a la red de comunicación. Es importante mencionar además que el switch y componentes de red están en una red Ethernet de 100Mbps con DHCP habilitado lo que hace a la red muy compatible con otros equipos. Para que la red funcione correctamente una PC estará gestionando las Raspberry Pi por el protocolo SSH puerto 22, de esta manera se podrá comandar estas para que realicen el envío de datos. En el capítulo 5 se tratarán a detalle todos los puntos que hacen referencia a las pruebas de la red de comunicación con el estándar DDS.

4.2.1 Adquisición de datos de frecuencia de la red eléctrica.

Para realizar la adquisición de datos de frecuencia es importante tener presente que no es posible medir estos valores directamente con la Raspberry Pi ya que la red eléctrica trabaja a un voltaje de 220 voltios y en este caso no buscamos medir el voltaje sino la frecuencia por lo que es

necesario diseñar y construir un circuito electrónico que permita mandar las señal analógica de frecuencia a los pines de entrada de la Raspberry Pi. El voltaje de la línea eléctrica que llega a los hogares es una señal analógica y para poder medir su frecuencia existe varios métodos entre los cuales están:

- ❖ utilizar un transformador de voltaje + una tarjeta electrónica.
- ❖ utilizar un sensor de corriente + una tarjeta electrónica.

En esta oportunidad se dispondrá de un transformador pequeño que tendrá la tarea de reducir el voltaje de la línea eléctrica de un voltaje de 220v (voltaje en la red eléctrica) a un valor de 6v de corriente alterna para luego pasar esta señal a la tarjeta electrónica la cual tiene un sistema que permite regular de mejor manera el voltaje y acoplarlo para luego la salida de esta tarjeta pasarla al conversor analógico digital de la Raspberry Pi para procesarla.

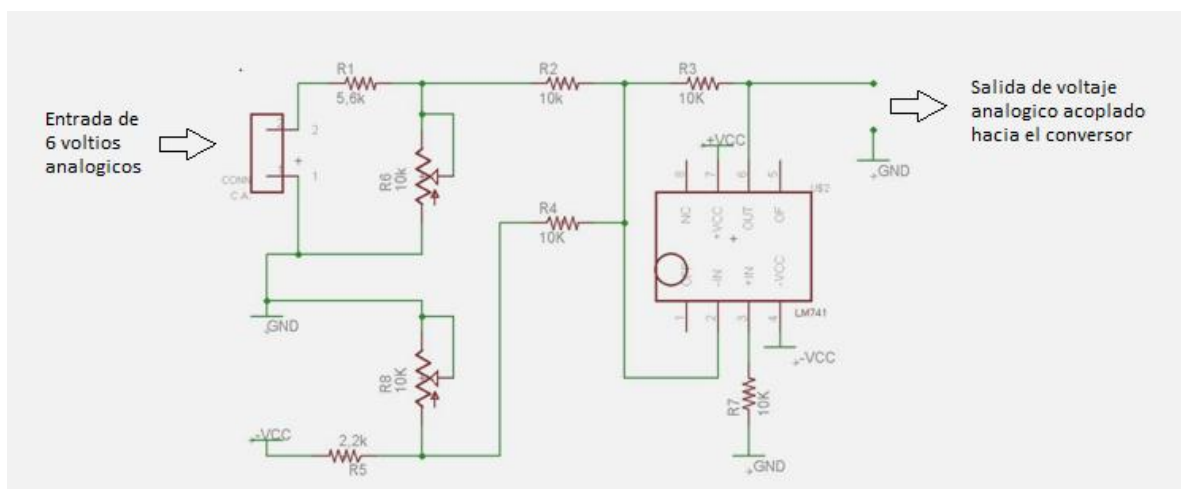


Figura 14: Esquema electrónico de la placa intermedia para la recolección de datos de frecuencia.
Fuente: Propia

Como se puede ver en figura 14 los componentes de la tarjeta electrónica son pocos y fáciles de conseguir, además el circuito es relativamente pequeño. La utilidad de la placa es preparar la señal que sale del transformador para alimentar el convertidor analógico-digital que manda los datos a su vez a la Raspberry Pi. A continuación se mostrará la manera correcta de armar el esquema electrónico incorporando el transformador, los adaptadores, la placa electrónica, el conversor A –D y la Raspberry Pi.

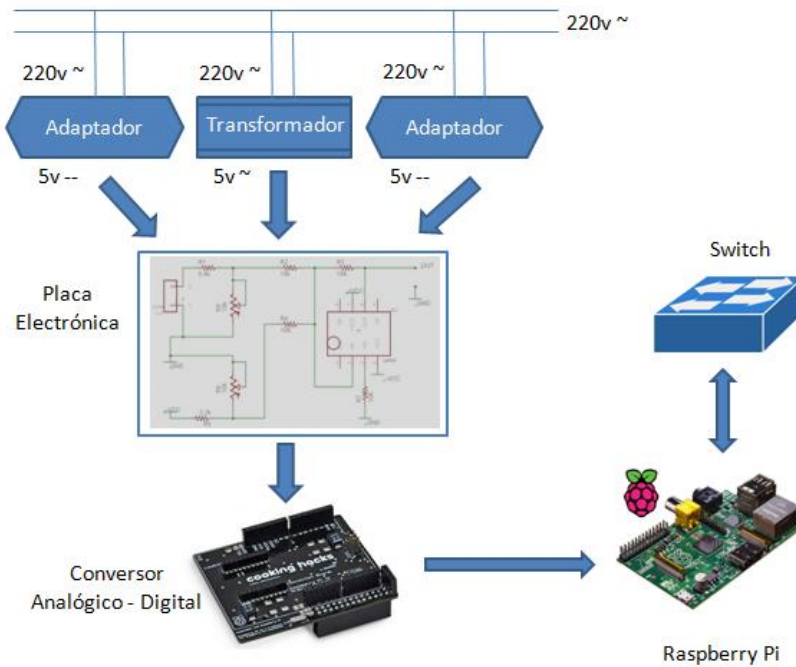


Figura 15: Esquema electrónico para armar el circuito de recolección de datos de frecuencia.
Fuente: Propia

Como se puede ver la placa electrónica necesita tanto un transformador para disminuir la señal analógica y dos adaptadores para la respectiva alimentación de la tarjeta. Una vez que la placa esté alimentada y recibiendo la señal correspondiente, dará una salida, la cual será conectada al convertor analógico digital montado en la Raspberry Pi.

4.2.2 Programa C++ en Raspberry Pi para la recolección de datos de frecuencia.

El programa para el cálculo de la frecuencia está realizado con el lenguaje de programación C++ que brinda las facilidades necesarias y que permite ser compilado y ejecutado desde la plataforma Raspbian del SBC (Raspberry Pi). La metodología a utilizar consta de dos partes importantes que son:

1. Adquisición de datos digitales desde el convertor analógico digital de la Raspberry Pi.
2. Cálculo y presentación de datos de frecuencia.

Para la adquisición de datos es necesario instalar la librería ArduPi en el SBC la cual sirve para que el programa reconozca la tarjeta electrónica del convertor. La librería está disponible en la web [27], así como también un instructivo de cómo montar y programar las primeras líneas de código con el propósito de configurar las entradas y salidas del convertor. En la parte del cálculo de la frecuencia se dispone de un cronómetro que servirá para calcular el tiempo en el que la onda sinusoidal cumple un ciclo completo, de esta manera se tendrá el periodo (T) el cual es un dato muy relevante en el cálculo de la frecuencia debido a que:

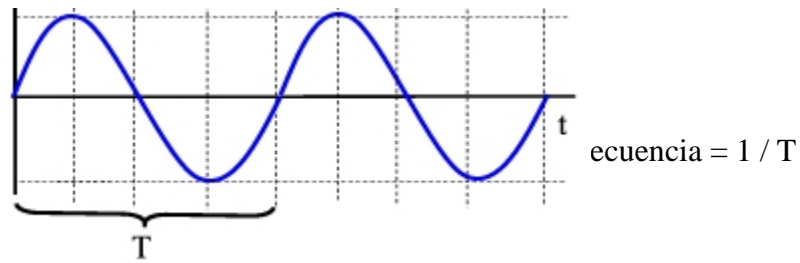


Figura 16: Periodo de la señal T y cálculo de la frecuencia. Fuente: www.fisicapractica.com

Con esta fórmula se calcula la frecuencia en Hertz directamente y se muestra en pantalla los resultados calculados al tomar el tiempo de un ciclo e implementar la fórmula de frecuencia. De esta manera se puede obtener de una forma sencilla la frecuencia. Cabe señalar que la señal analógica será muestreada por el conversor analógico digital de una forma continua.

4.2.3 Configuración de DDS para enviar datos de frecuencia.

El propósito principal de la instalación de DDS con todos sus ejemplos es poder recoger datos de frecuencia con el SBC Raspberry Pi y enviar estos por medio de DDS y sus ejecutables HelloPublisher y HelloSuscriber a la red comunicación. Para lograr enviar estos datos recogidos se pretende cambiar algunas líneas de código HelloPublisher de DDS con el objetivo de tomar datos del programa (cálculo de la frecuencia que funciona en la Raspberry Pi) y enviarlos por DDS.

Antes de empezar a realizar las configuraciones necesarias en DDS es recomendable probar que el programa de cálculo de la frecuencia esté funcionando correctamente. Las pruebas de esta parte del sistema se podrán observar en el siguiente capítulo relacionado a resultados y discusión.

Una vez que el programa de frecuencia esté funcionando correctamente se pretende comenzar a realizar unos pequeños cambios en el código de DDS específicamente en el código denominado HelloPublisher.c ubicado en:

`/home/pi/RTI/ndds.5.1.0/example/C/Hello_simple/src/`

A continuación se digita el siguiente código para visualizar el archivo HelloPublisher.c

`nano HelloPublisher.c`

Las figuras 17 y 18 muestran la parte del contenido original del código y la parte del contenido modificado del código HelloPublisher.c para explicar de mejor manera las modificaciones realizadas.

```

for (;;) {
    printf("Please type a message> ");
    if (fgets(sample, MAX_STRING_SIZE-1, stdin) == NULL) {
        break;
    }
    /* Removes the newline from input string */
    sample[strlen(sample)-1] = '\0';
    retcode = DDS_StringDataWriter_write(
        string_writer,
        sample,
        &DDS_HANDLE_NIL);
    if (retcode != DDS_RETCODE_OK) {
        printf("Write failed: %d\n", retcode);
    }
    if (strlen(sample) == 0) {
        break;
    }
}

```

Figura 17: Parte del código original del programa HelloPublisher.c de DDS. Fuente: Propia

```

for (;;) {
    //printf("Please type a message> ");
    if (fgets(sample, MAX_STRING_SIZE-1, stdin) == NULL) {
        break;
    }
    /* Removes the newline from input string */
    //sample[strlen(sample)-1] = '\0';
    fichero = popen ("sudo ./f/freq3", "r");
    fgets (aux, 1000, fichero);
    while (!feof (fichero))
    {
        //strcpy(aux, "Diego");
        strcat(sample, aux);
        retcode = DDS_StringDataWriter_write(
            string_writer,
            sample,
            &DDS_HANDLE_NIL);
        fgets (aux, 1000, fichero);
        sample[0]='\0';
    }
    if (retcode != DDS_RETCODE_OK) {
        printf("Write failed: %d\n", retcode);
    }
    if (strlen(sample) == 0) {
        break;
    }
    //
    // aux[0]='\0';
    // sample[0]='\0';
    pclose (fichero);
}

```

Figura 18: Parte del código modificado del programa HelloPublisher.c de DDS. Fuente: Propia

La parte del código que se indica en las dos figuras anteriores es la más importante del programa ya que es la encargada de enviar los datos por DDS. En el código se puede ver la línea con la herramienta "popen" la cual llama al ejecutable del programa freq4 para extraer los datos de frecuencia que luego serán almacenados en la variable aux que se encuentra en la siguiente línea. Posteriormente usa otra herramienta que se denomina strcat la cual concatena el valor de aux en la variable sample, esta es necesaria ya que el programa HelloPublisher envía por la red la variable sample la cual viaja por la red encapsulada en el protocolo RTPS2 (Real-Time Publisher Subscriber 2) de DDS y bajo un nombre de tópicos. El dato viaja por la red y es recibido por el terminal suscriptor asociado.

Popen es una herramienta poderosa de programación y sirve para capturar datos de salida de un programa y llevarlos a otro programa como datos de entrada. Está basado en Pipes que son

tuberías de programación las cuales sirven para intercambiar datos de un programa a otro. En la transmisión de la frecuencia, popen servirá para pasar los datos de frecuencia del programa del cálculo de frecuencia (freq4.c) al programa que enviara los datos a la red por DDS (HelloPublisher).

Como se indico en puntos anteriores cada dato de DDS posee un nombre o tópicos que puede ser dado por el administrador. Por ejemplo, si es necesario enviar temperatura, el administrador puede mandar los datos de grados Celsius bajo el tópicos “temperatura1” y de esta misma forma se deben transferir los datos deseados. El terminal receptor debe tener ejecutado el programa HelloSuscriber con el mismo tópicos “temperatura1” para recibir el dato de una forma correcta, caso contrario no podrá recibir la temperatura. Existe una parte en el código tanto en el programa HelloPublisher como en el programa HelloSuscriber que hace referencia a este punto de los tópicos, dicha parte será indicada a continuación.

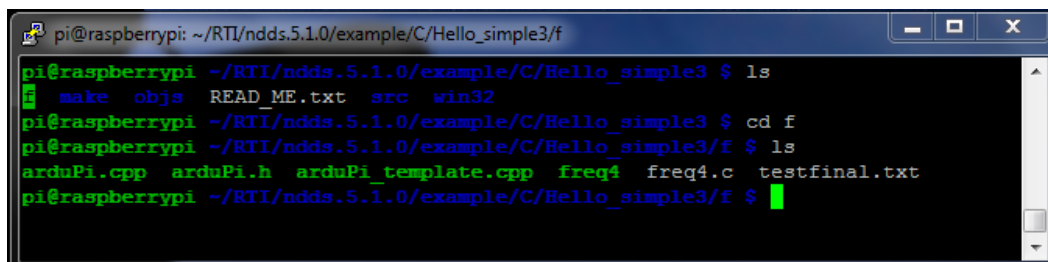
```
/* Create the topic "Hello, World" for the String type */
topic = DDS_DomainParticipant_create_topic(
    participant,
    "Hello, World", /* Topic name*/
    DDS_StringTypeSupport_get_type_name(), /* Type name */
    &DDS_TOPIC_QOS_DEFAULT, /* Topic QoS */
    NULL, /* Listener */
    DDS_STATUS_MASK_NONE);

if (topic == NULL) {
    puts("Unable to create topic.");
    goto clean_exit;
}
```

Figura 19: Parte del código en donde está el tópicos asociado. Fuente: Propia

Como se puede apreciar en el ejemplo de la figura 19 el tópicos asociado es “Hello, World” de esta manera todos los terminales que tengan asociado este tópicos podrán introducir datos a este o leer datos de este. De esta forma se puede clasificar de cierta manera los datos que están viajando por la red de comunicación. Cada dato numérico de frecuencia (por ejemplo 49,85Hz) viaja por la red bajo un tópicos En las pruebas del capítulo siguiente se usaran dos nombres de tópicos distintos los cuales son: Hello, World y Hello, mundo.

Para que las configuraciones estén bien realizadas se requiere que el programa freq4.c este bajo el directorio correspondiente, esto es debido a que el programa HelloPublisher3 llamara a freq4.c ubicado en una determinada zona. La figura 20 muestra algunos de los directorios importantes a considerar.



```
pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple3/f
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple3 $ ls
make objs READ_ME.txt src win32
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple3 $ cd f
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple3/f $ ls
arduPi.cpp arduPi.h arduPi_template.cpp freq4 freq4.c testfinal.txt
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple3/f $
```

Figura 20: Directorio de archivos de Hello_simple3. Fuente: Propia.

Se observa en la figura 20 que bajo el directorio Hello_simple3 están algunas carpetas entre ellas una llamada “f” en donde se guardan todos los archivos que pertenecen al programa que calcula la frecuencia freq4.c, esta carpeta incluye las librerías ArduPi, el programa propiamente y su ejecutable. Una vez hechas todas estas configuraciones se deberá compilar el programa Hello_simple3 de la misma manera que se indica en el anexo B, es decir se usará el comando make y su archivo correspondiente para lograr una compilación correcta para luego ejecutar el programa HelloPublisher en la Raspberry Pi y el HelloSubscriber en la Pc o las PCs.

4.2.4 Consideraciones importantes en la red.

Cuando la red esté completamente armada y cada dispositivo en la red (PCs, Raspberry Pi 1 y 2) tenga instalado y configurado DDS es preciso saber cómo funciona la red de comunicación DDS para el envío de datos y cuáles son las conexiones y ejecuciones antes del envío de datos. Para resolver estas dudas se empezará definiendo cada programa que se ejecutara en cada dispositivo, de esta manera se tendrá una idea más clara de cómo poner en funcionamiento la red de transmisión de frecuencia por DDS. Los elementos presentes en la red son 5, a continuación se especifica cada uno de estos y se presentan sus cualidades.

Computador 1 (PC_1)

- Computador HP con sistema operativo Centos 6.5.
- DDS DE RTI para Red Hat Enterprise Linux 6.
- Hello_simple con tópico Hello, World.
- Hello_simple1 con tópico Hello, mundo.

Raspberry Pi 1 (RP_1)

- Raspberry Pi modelo B con sistema operativo 2014-06-20-wheezy-raspbian.
- DDS DE RTI Connex DDS Professional-5.1.0.
- Hello_simple3 con tópico Hello, World.
- Software para el cálculo de frecuencia freq4.c

Computador 2 (PC_2)

- Toshiba Satellite con RTI Connex DDS Linux Live ISO (con RTI Connex DDS Professional).
- Hello_simple con tópico Hello, World.
- Hello_simple1 con tópico Hello, mundo.

Raspberry Pi 2 (RB_2)

- Raspberry Pi modelo B con sistema operativo 2014-06-20-wheezy-raspbian.
- DDS DE RTI Connex DDS Professional-5.1.0.
- Hello_simple3 con tópico Hello, mundo.
- Software para el cálculo de frecuencia freq4.c

Computador 3 (PC_3)

- computador HP con sistema operativo Centos 6.5.
- DDS DE RTI para Red Hat Enterprise Linux 6.
- Hello_simple con tópico Hello, World.

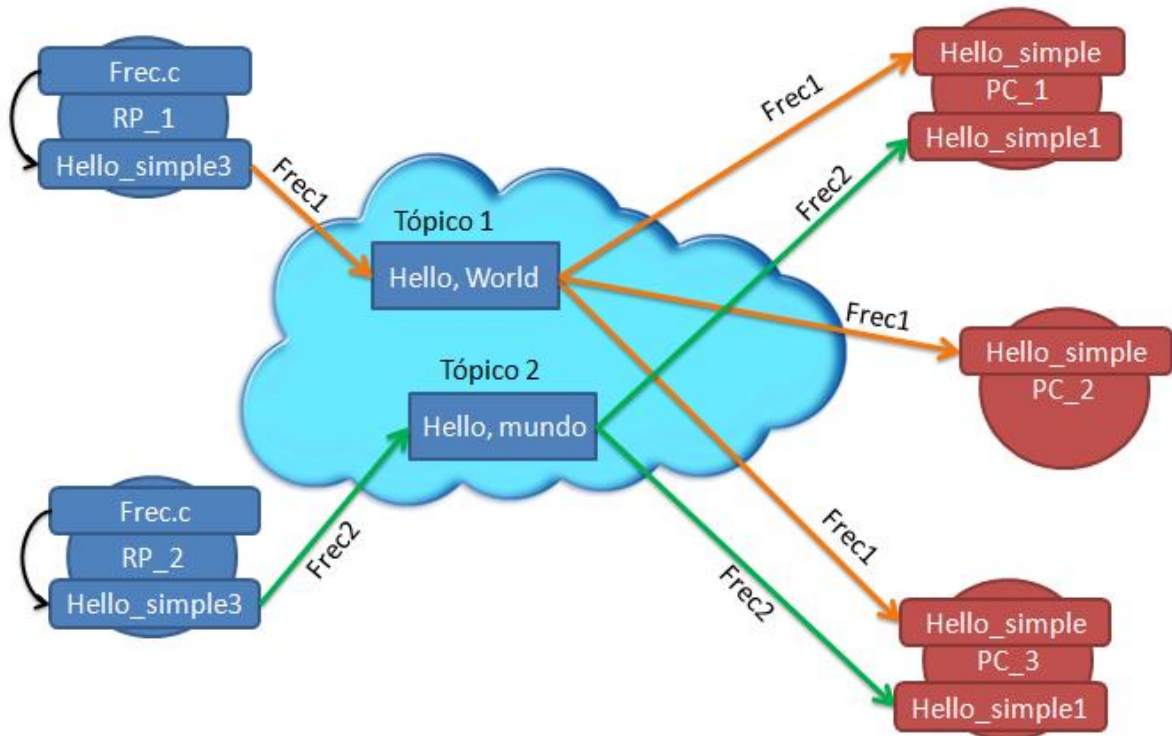


Figura 21: Elementos de la red y tópicos asociados. Fuente: Propia.

Como se puede observar en la figura 21, cada frecuencia viajará del publicador al suscriptor por la red con su nombre de tópico correspondiente. Aquel suscriptor que desee recibir determinada frecuencia debe ser ejecutado con el tópico en cuestión, es decir por ejemplo si pretendo enviar datos de frecuencia con la Raspberry Pi 1 y recibir con el computador 1 si es posible el intercambio. Para que esta transmisión sea efectiva se debe ejecutar el HelloPublisher (con tópico Hello_World) del ejemplo Hello_simple3 en la Raspberry Pi 1 y ejecutar el HelloSuscriber (con tópico Hello, World) del ejemplo Hello_simple en la Pc 1. De esta forma se estarán transmitiendo los datos de frecuencia de forma correcta. Si por ejemplo deseamos recibir esta misma frecuencia con el computador 2 si es posible, tan solo ejecutamos el HelloSuscriber (con tópico Hello, World) del ejemplo Hello_simple en la Pc 2.

Con lo dicho anteriormente se puede empezar las pruebas de transmisión con el estándar DDS y se podrá conocer en el siguiente capítulo de una manera más profunda en qué consisten estas pruebas y su respectivo análisis, así como también la comparación de este estándar con otra aplicación TCP y observar cada uno de los comportamientos al enviar y recibir los datos.

5 CAPITULO 5 RESULTADOS Y DISCUSIÓN

En este capítulo se describen e ilustran los resultados de las diferentes pruebas que se realizaron en el transcurso de todo el proceso de programación, instalación y configuración de los distintos componentes para llevar a cabo la transmisión y recepción de la frecuencia de la línea eléctrica con el estándar DDS. Además se usara los computadores normales para la recolección de datos y los SBCs o Raspberry Pi para la adquisición y transmisión de la frecuencia como tal. Cabe recordar que cada prueba tendrá una sección inmediata inferior acerca de la discusión.

5.1 Pruebas y resultados del programa que calcula la frecuencia.

Antes de realizar las pruebas de envío y recepción de datos de frecuencia por la red con el estándar DDS, es necesario asegurar que el programa que recoge la frecuencia esté funcionando correctamente. Para esto se dispondrá de un generador de funciones que pretende simular la señal de energía eléctrica de la red y que permite obtener diferentes frecuencias de una señal sinusoidal. La señal será ingresada al conversor para comprobar en pantalla los cambios que tienen los resultados al variar la perilla del generador de funciones. La siguiente figura muestra los resultados obtenidos comparando la frecuencia mostrada en pantalla del programa en C++ y los resultados del generador de funciones.



Figura 22: Datos de frecuencia del generador de funciones y del programa C++. Fuente: Propia

Como se puede ver en la figura 22 los datos calculados en el programa C++ presente en la Raspberry Pi corresponden a la frecuencia de la onda sinusoidal del generador de funciones que en este caso es de 50Hz. También se puede variar la frecuencia del generador y observar los resultados que imprime en pantalla el programa, de esta manera se asegura su buen funcionamiento. Las pruebas realizadas en laboratorio dieron como resultado que las frecuencias de la red eléctrica que son de 50Hz en unos países y 60 Hz en otros es medible. A continuación se calculara la desviación estándar y error estándar de la media de 585 muestras de datos de frecuencia recogidos.

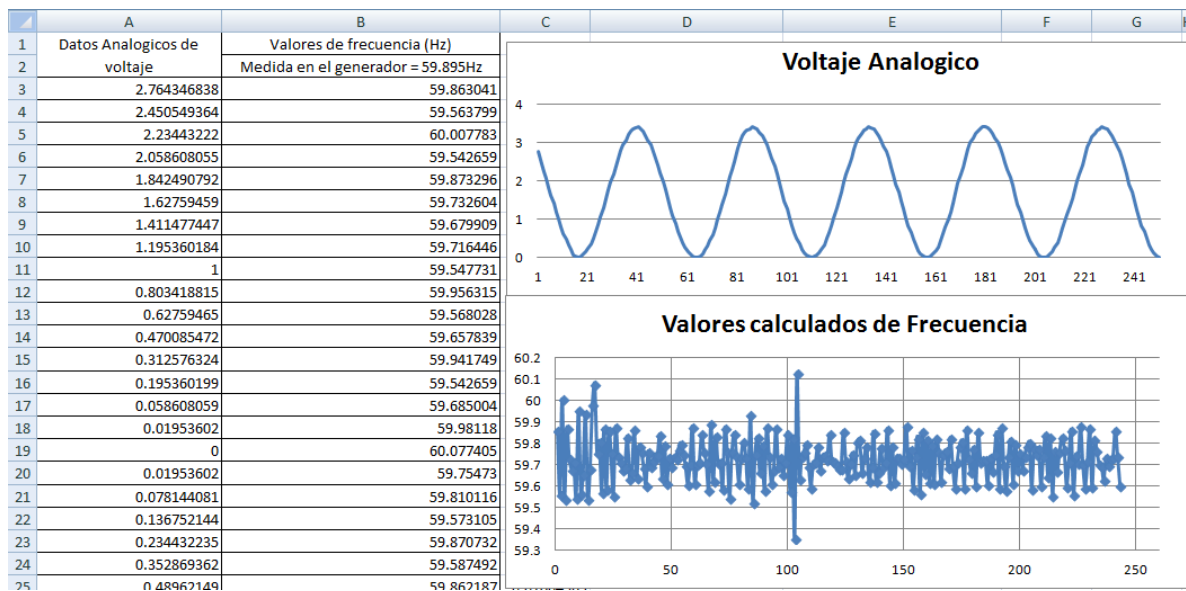


Figura 23: Cálculo de la varianza, desviación estándar y error estándar de la media. Fuente: Propia.

La figura 23 muestra en primer lugar los datos digitales muestreados (columna A) de la señal sinusoidal (voltaje analógico) que son de una onda que varía entre 0 y 4 voltios, la misma que es graficada como una señal sinusoidal y que se parece mucho a la señal de la red eléctrica. En segundo lugar se ven los valores de frecuencia (columna B) recogidos para el cálculo de la varianza, desviación estándar y error estándar. Estos datos son adquiridos de un generador de funciones como instrumento patrón de medida. Las fórmulas para el cálculo son las siguientes:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (X_i - m)^2$$

$$\text{Desv. Estándar} = \sqrt{\sigma^2} = \sigma$$

Donde:

n = número de muestras

X = cada dato

m = media o promedio

σ^2 = varianza

Luego

$$\text{Error Estándar de la media} = \frac{\sigma}{\sqrt{n}}$$

Como se indicó anteriormente, para tener una mejor apreciación de la frecuencia recogida, se calcula la desviación estándar de 585 muestras a 59,8Hz así como también el error estándar. Los resultados de este cálculo muestran que la desviación estándar es de 0,1068 y su error estándar de la media es de 0.00441. De este modo, el error en el resultado no es superior a ± 0.00441 (68% de

confianza) o no más de $2(0.00441) = \pm 0.00883$ (a un 95% de confianza). Finalmente en esta parte se podría calcular la tasa de muestreo del conversor analógico digital sobre las muestras del voltaje sinusoidal la cual está en un promedio de 2880 muestras por segundo. Estas formulas estadísticas utilizadas en ese punto son avaladas por la ASTM (American Society for Testing and Materials).

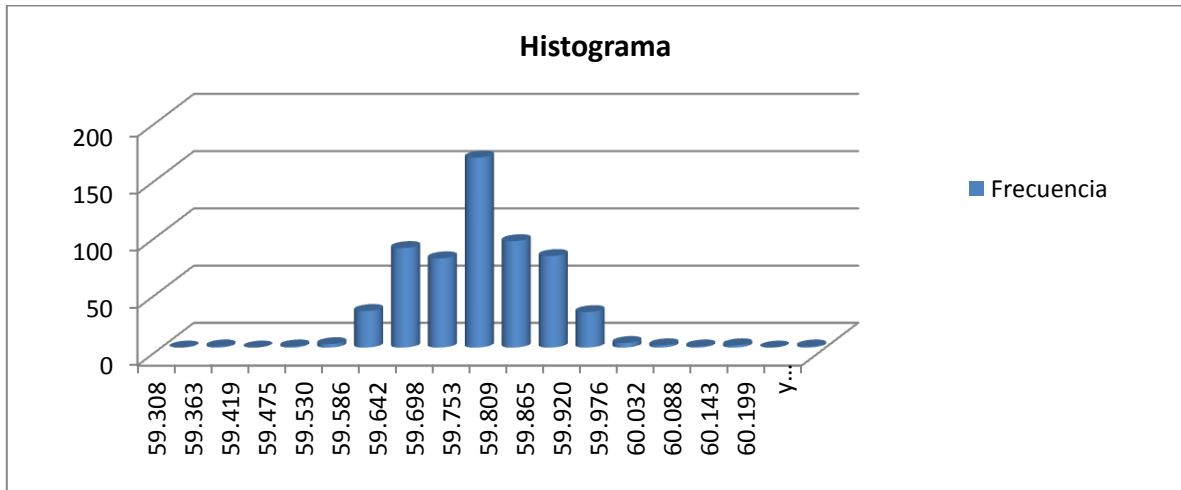


Figura 24: Histograma de los valores de frecuencia obtenidos desde la Raspberry Pi.

La figura 24 indica un análisis de datos de frecuencia y se caracteriza por recalcar los resultados que más se repiten en el conjunto de datos. Es sencillo predecir qué resultado se replica como es el caso de por ejemplo la frecuencia de valor 59.753 se repite 150 veces en el total de muestras y así se pueden ir comparando otras medidas. Es preciso recalcar que tal histograma fue realizado con el número total de 585 datos de frecuencia.

Como se mencionó anteriormente el programa C++ en el SBC para la recolección de datos de frecuencia, se basa en contabilizar el tiempo de un periodo y calcular la frecuencia. Esto se logra con la detección de la onda sinusoidal eléctrica que cruzar por un umbral fijo, de este modo se sabe cuando iniciar el cronómetro que mide el periodo. Esta técnica solo funciona con señales sinusoidales puras, de esta forma si se tienen ondas con contenido armónico esta lectura puede entregar mala información. Por esta razón se realizó otra prueba para el cálculo de la frecuencia en donde se involucra un filtro pasa bajo. Este filtro puede eliminar parte del ruido producido por contenido armónico y ayudar a tener mejores resultados en el cálculo de la frecuencia.

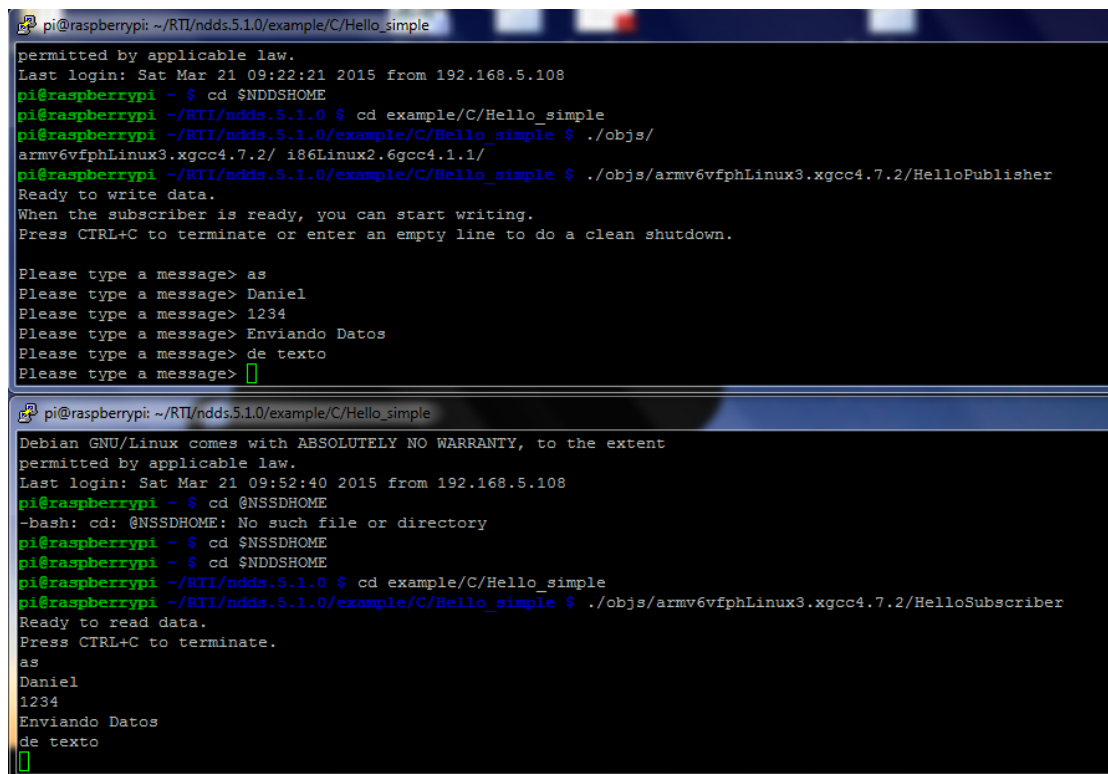
Discusión

Las pruebas de medida de frecuencia dan como resultado una desviación de 0,1 es decir los datos no se encuentran muy dispersos. Debido a que la frecuencia en una medida usada en micro-redes es necesaria tener tal dispersión e incluso menos, ya que es una medida que indica la variación de lo generado con respecto a lo consumido en una red eléctrica. Esta medida debe estar lo más estable posible para un buen funcionamiento y estabilidad de la red en campo. La medida de la frecuencia realizada con el programa en C++ puede medir frecuencias mayores ya que el

Raspberry Pi es capaz de muestrear a gran velocidad y recoger la cantidad de muestras suficientes para mostrar resultados de frecuencias mayores pero la desviación también crece.

5.2 Pruebas y resultados del envío de datos simples por DDS

Luego de verificar el funcionamiento del cálculo de la frecuencia es necesario realizar las primeras pruebas para apreciar el buen funcionamiento de DDS y sus ejemplos. En esta parte se espera enviar y recibir texto simple entre el publicador y el suscriptor que pueden o no estar en la misma PC. En esta ocasión se dispondrá de tres computadores para esta prueba con el DDS instalado y configurado según el anexo A. Se recuerda que los tres computadores deben estar en la misma red y tener direcciones IP definidas.



```
pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple
permitted by applicable law.
Last login: Sat Mar 21 09:22:21 2015 from 192.168.5.108
pi@raspberrypi ~ $ cd $NDDSHOME
pi@raspberrypi ~/RTI/ndds.5.1.0 $ cd example/C/Hello_simple
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple $ ./objs/
armv6vfpLinux3.xgcc4.7.2/ i86Linux2.6gcc4.1.1/
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple $ ./objs/armv6vfpLinux3.xgcc4.7.2/HelloPublisher
Ready to write data.
When the subscriber is ready, you can start writing.
Press CTRL+C to terminate or enter an empty line to do a clean shutdown.

Please type a message> as
Please type a message> Daniel
Please type a message> 1234
Please type a message> Enviando Datos
Please type a message> de texto
Please type a message>

pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 21 09:52:40 2015 from 192.168.5.108
pi@raspberrypi ~ $ cd @NSSDHOME
-bash: cd: @NSSDHOME: No such file or directory
pi@raspberrypi ~ $ cd $NSSDHOME
pi@raspberrypi ~ $ cd $NDDSHOME
pi@raspberrypi ~/RTI/ndds.5.1.0 $ cd example/C/Hello_simple
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple $ ./objs/armv6vfpLinux3.xgcc4.7.2/HelloSubscriber
Ready to read data.
Press CTRL+C to terminate.
as
Daniel
1234
Enviando Datos
de texto

```

Figura 25: Pruebas de transmisión de texto simple con DDS. Fuente: Propia.

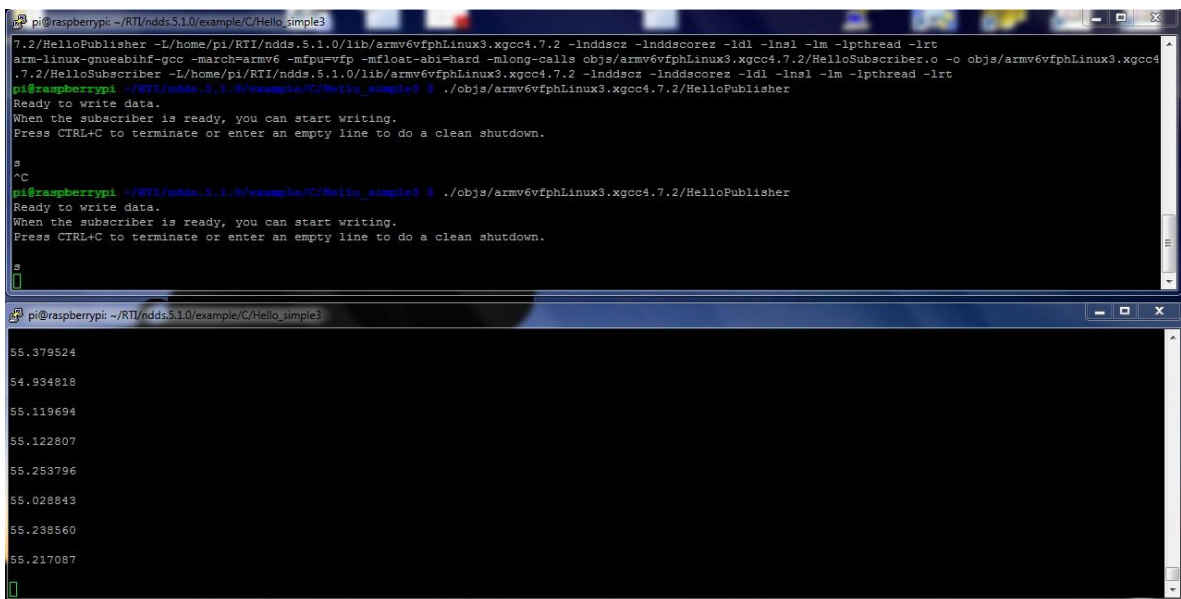
Como se puede ver en la figura 25 existen para esta prueba dos terminales, el superior se está corriendo el publicador y el terminal inferior está ejecutando el suscriptor para la transmisión de datos a través del estándar DDS. Las pruebas de esta parte muestran que cualquier dato simple de texto o numérico escrito desde un teclado en el publicador será enviado al suscriptor y recibido por este como se muestra, de esta manera se puede comprobar la funcionalidad de las primeras pruebas del estándar. Es preciso recordar que esta prueba se realizó entre tres computadores obteniendo buenos resultados en el envío de datos.

Discusión.

En esta prueba de la transmisión de datos simples se puede acotar que los ejemplos que trae DDS son adecuados para pruebas simples tan solo de transmisión de textos o números sin que sean asociados a ninguna variable o parámetro de medida. Estos ejemplos pueden ser manipulados directamente desde un editor de texto para enviar otro tipo de dato más elaborado asociado a algún proceso o medida. Esto es justo lo que se pretende realizar en las próximas pruebas de tal manera de tomar datos verdaderos de una fuente de medición o control para luego llevarlos a los ejemplos de DDS y enviarlos a la red con el propósito de realizar pruebas más elaboradas de este estándar.

5.3 Pruebas y resultados del envío de una sola frecuencia por DDS.

La siguiente prueba es la transmisión de datos de frecuencia por la red sobre el estándar, para esto se configura y se programa tanto los computadores como las Raspberry Pi con el estándar DDS y el programa freq4.c corriendo en el SBC. En esta parte se dispondrá de publicadores en la Raspberry Pi que se encuentran bajo el directorio Hello_simple3 y suscriptores en las PCs que se encuentran en el directorio Hello_simple para que de esta manera exista un correcto envío de datos. Todos los elementos que deseen enviar o recibir datos deben estar en la misma red.



```
pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple3
7.2/HelloPublisher -L/home/pi/RTI/ndds.5.1.0/lib/armv6vfpLinux3.xgcc4.7.2 -lnddscz -lnddscorz -ldl -lnsl -lm -lpthread -lrt
arm-linux-gnueabihf-gcc -march=armv6 -mcpu=vfp -mfloat-abi=hard -mlong-calls obj/armv6vfpLinux3.xgcc4.7.2/HelloSubscriber.o -o obj/armv6vfpLinux3.xgcc4
7.2/HelloSubscriber -L/home/pi/RTI/ndds.5.1.0/lib/armv6vfpLinux3.xgcc4.7.2 -lnddscz -lnddscorz -ldl -lnsl -lm -lpthread -lrt
pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple3 ./obj/armv6vfpLinux3.xgcc4.7.2/HelloPublisher
Ready to write data.
When the subscriber is ready, you can start writing.
Press CTRL+C to terminate or enter an empty line to do a clean shutdown.
s
^C
pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple3 ./obj/armv6vfpLinux3.xgcc4.7.2/HelloSubscriber
Ready to write data.
When the subscriber is ready, you can start writing.
Press CTRL+C to terminate or enter an empty line to do a clean shutdown.
s
55.379524
54.934818
55.119694
55.122807
55.253796
55.028843
55.238560
55.217087
```

Figura 26: Prueba de transmisión de datos reales de frecuencia con DDS. Fuente: Propia

La figura 26 muestra el envío y recepción de datos de frecuencia de 55Hz en la misma Raspberry Pi lo cual si es posible hacerlo para comprobar el correcto funcionamiento de publicadores y suscriptores. De la misma manera que en la prueba anterior, el terminal superior está corriendo el publicador y el inferior está ejecutando el suscriptor por esto es que se pueden apreciar datos de frecuencia recibidos en la figura. Es oportuno recordad que estas pruebas también son posibles hacerlas desde un equipo a otro en la misma red siempre que un elemento funcione como publicador y el otro funcione como suscriptor y existan datos a transmitir por la red.

Discusión.

El envío de la frecuencia por la red de comunicación es un método en el cual el programa HelloPublisher.c recoge los datos del programa freq.c que calcula tal frecuencia para transmitir. Este proceso se podría mejorar con la incorporación directa del cálculo de la frecuencia en el mismo DDS. Se intentó realizar esto pero no tuvo éxito debido a un problema de librerías y ajustes de software. También es preciso destacar que los datos se envían inmediatamente es decir, frecuencia que se calcula, frecuencia que se envía de forma inmediata, no se conoce con certeza el tiempo que se demora entre tener el dato y mandar a la red pero es el tiempo en el que se demora en recorrer unas pocas líneas de código. El envío de estos datos se realiza desde el publicador que en este caso es la Raspberry Pi hacia los publicadores que son las PCs.

5.4 Pruebas en el manejo de Tópicos.

En el capítulo anterior se explicó lo que son los tópicos y cuál es su función en el estándar. Los ejemplos que trae DDS por omisión cuentan con el tópico “Hello, World” tanto publicadores como suscriptores, esto quiere decir que si varios publicadores envían diferentes tipos de datos como por ejemplo temperatura, humedad y velocidad, todos los suscriptores recibirán estos mismos datos pero maclados, esto se debe a que todos los datos no fueron previamente configurados con un nombre de tópico respectivo. Para las siguientes pruebas hay que considerar el manejo de tópicos ya que se involucrará otra frecuencia diferente y una segunda Raspberry Pi con esto se enviarán por la red dos datos diferentes. Si el nombre de tópico de la frecuencia 1 no es diferente al tópico de la frecuencia 2 todos los terminales suscriptores conectados a la red recibirán las dos frecuencias mezcladas. Es por esto que se dispondrá del tópico “Hello, World” para la transmisión de la primera frecuencia y “Hello, mundo” para la transmisión de la segunda frecuencia, de esta forma se podrá tener terminales suscriptores con cada nombre de tópico para recibir la frecuencia deseada.

```

pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple3
Ready to write data.
When the subscriber is ready, you can start writing.
Press CTRL+C to terminate or enter an empty line to do a clean shutdown.
^V^C
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple3 $ ./objs/armv6vfpLinux3.xgcc4.7.2/HelloPublisher
Ready to write data.
When the subscriber is ready, you can start writing.
Press CTRL+C to terminate or enter an empty line to do a clean shutdown.
PUBLICADOR 1

pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple3
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple3 $ ./objs/armv6vfpLinux3.xgcc4.7.2/HelloSubscriber
Ready to read data.
Press CTRL+C to terminate.
^[[A^C^C
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple3 $ ./objs/armv6vfpLinux3.xgcc4.7.2/HelloPublisher
Ready to write data.
When the subscriber is ready, you can start writing.
Press CTRL+C to terminate or enter an empty line to do a clean shutdown.
PUBLICADOR 2

pi@raspberrypi: ~/RTI/ndds.5.1.0/scripts
a.00)
Copyright 2012 Real-Time Innovations, Inc.
~~~~~
NddsSpy is listening for data, press CTRL+C to stop it.
NDDSSPY

source_timestamp  Info  Src HostId  topic  type
-----
1413953906.510663  [W] +N  C0A80510  Hello, World  DDS::String  PUBLICADOR 1
1426936287.126193  [W] +N  C0A80519  Hello, mundo  DDS::String  PUBLICADOR 2

```

Figura 27: Publicadores con diferente t3pico en DDS. Fuente: Propia.

En la figura 27 se muestra a dos publicadores listos para mandar los datos de frecuencia, cada uno con diferente nombre de t3pico l3gicamente asociado a cada frecuencia. Adem3s se puede ver que en el tercer terminal de la figura se encuentra ejecutando un programa ayuda de DDS llamado “nddsspy”, el cual realiza un escaneo por la red y trata de encontrar los publicadores y suscriptores ejecutados en la red y tambi3n el nombre de t3pico asociado a cada uno. Este software se encuentra en el directorio /home/pi/RTI/ndds5.1.0/scripts/. Por ejemplo si queremos enviar un dato con t3pico “velocidad” los 3nicos suscriptores que podr3n recibir estos ser3n los que est3n asociados con el mismo t3pico. Hay que recordar que para esta prueba el publicador uno esta ejecutado en la Raspberry Pi 1, el publicador dos en las Raspberry Pi 2 y el nddsspy podr3 ser ejecutado en cualquier terminal conectado a la red.

A continuaci3n se mostrar3 c3mo se puede ejecutar tanto publicadores como suscriptores y observar lo que nddsspy muestra en pantalla al ser ejecutado.

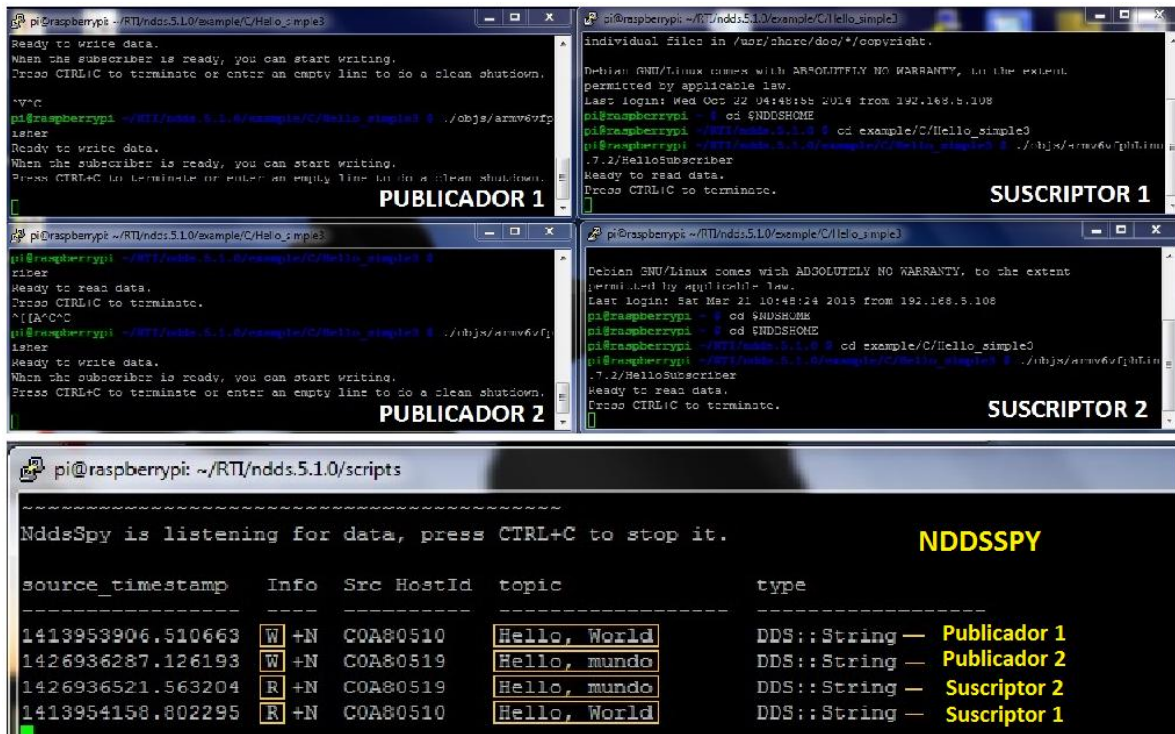


Figura 28: Dos publicadores y dos suscriptores con diferentes. Fuente: Propia.

La figura 28 muestra 5 terminales en donde se especifica la función de cada uno, también es importante ver que nddsspy recogió datos de los elementos que se han ejecutado en la red que en este caso son dos publicadores y dos suscriptores con sus respectivos nombres de tópico. Esto hará posible enviar dos frecuencias distintas sin mezclar los datos.

Discusión.

El Software de DDS denominado nddsspy es de mucha ayuda para analizar los publicadores y suscriptores presentes en la red, además permite ver los tópicos asociados a cada elemento de la red brindando información relevante que es aprovechada por el administrador. Es oportuno recordar que este software muestra información tanto en la activación de algún componente de red DDS como en el envío de datos. Además los nombres de tópico aparecen directamente sin tener que ingresar en el código facilitando la administración y configuración de datos por la red.

5.5 Pruebas y resultados del envío de dos diferentes frecuencias por DDS.

Luego de ver el manejo de diferentes tópicos en la red de comunicación se realizará una prueba más con la adición de otra frecuencia para ver el comportamiento de estas dos fuentes de datos en la red de comunicación. Con la ayuda de una primer SBC que recoge datos de frecuencia de la red eléctrica y una segundo Raspberry Pi que adquiere datos de un generador, se establecen las fuentes de las señales. Estos elementos serán configurados como publicadores en la transmisión teniendo cada una un tópico diferente para separar los datos como se vio en el punto anterior referente al manejo de tópicos. Las pruebas dan como resultado la emisión y recepción de las dos

frecuencias con fuentes y valores distintos. En la siguiente figura se podrá apreciar de mejor manera lo indicado anteriormente.

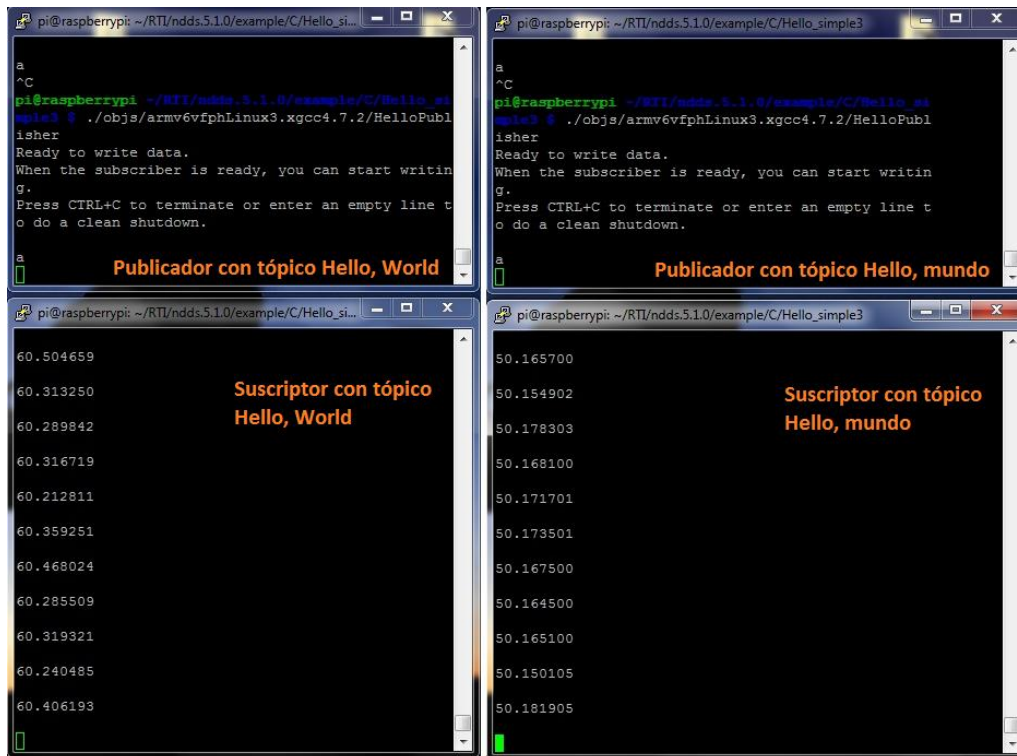


Figura 29: Envío y recepción por DDS de dos frecuencias. Fuente: Propia

La figura 29 muestra efectivamente la transmisión de dos diferentes datos de frecuencia uno correspondiente a los 60,2Hz y otro a los 50,1Hz. Esto se logra con la ayuda de dos fuentes la primera que es la medida de la fuente de generador de señales y la segunda es la medida de la frecuencia de red eléctrica convencional.

Discusión.

La parte más importante en el envío de datos de diferentes fuentes (temperatura, presión, frecuencia, voltaje, etc.) por la red de comunicación DDS es el tópico relacionado a cada tipo de dato ya que gracias a esto el suscriptor podrá ser configurado para recibir los datos que le correspondan. Si en la red existiera un solo tópico y varios tipos de datos con varios publicadores y suscriptores los datos se mezclarían, es decir todos los datos serian escritos por los publicadores en un mismo tópico y los suscriptores recibirían todos los datos desde este tópico llegando a tener una combinación de datos sin una clasificación clara de medidas. Es preciso recordar que en este caso se envían dos frecuencias con su tópico correspondiente y los suscriptores pueden recoger estos datos tan solo asociándose al tópico en cuestión.

5.6 Análisis de paquetes DDS de frecuencia.

Una vez que los datos de frecuencia están viajando por la red es posible realizar la captura de los paquetes para observar las características de estos, ver su velocidad y su contenido para de esta

manera saber cómo viajan los resultados de frecuencia por la red. Los paquetes capturados en la transmisión de datos por DDS son recogidos con la herramienta de líneas de comando tcpdump la cual es utilizada principalmente para el análisis de tráfico. En las pruebas de recolección de datos se usara tal herramienta que permite capturar datos en tiempo real y brinda la opción de guardar estos en un documento con un formato .pcap con el propósito de poder disponer de estos datos en el programa Wireshark que nos facilita ver de forma más simple y ordenada los datos de tráfico recogidos por tcpdump.

5.6.1 Paquetes recogidos desde un publicador o suscriptor

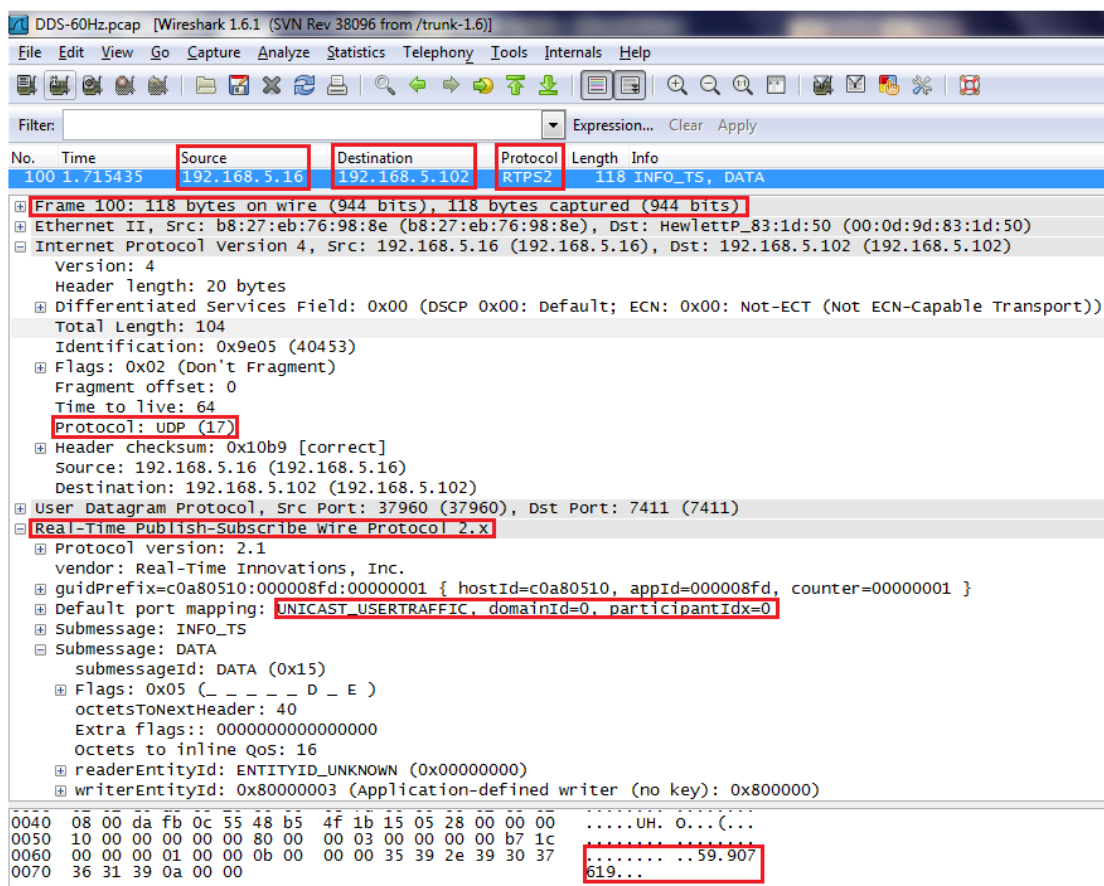


Figura 30: Paquete 100 de transmisión de frecuencia de 60Hz por DDS visto desde un suscriptor. Fuente: Propia.

Como se puede ver en la figura 30, el paquete de frecuencia DDS tiene algunas características como por ejemplo posee un tamaño de 118 bytes, también se observa que este maneja un protocolo UDP y es encapsulado por el protocolo propietario de DDS denominado RTPS2. Otro dato interesante es que el paquete muestra el número entero que lleva en su interior, este hace referencia a la medida de la frecuencia en cuestión. En este caso se analiza el paquete numero 100, ya que esta información se repite a lo largo de la transferencia de todos los paquetes involucrados.

Discusión.

El análisis de paquetes en las pruebas de transmisión dan como resultado que la red y estándar DDS es capaz de mandar por la red todos los datos recolectados en el cálculo y envío de frecuencia, es decir, la Raspberry Pi cuando termina de calcular un dato enseguida lo envía por la red de comunicación y en el otro extremo es recogida por el suscriptor asociado a tal dato. En la figura del análisis de paquete se observa que el paquete es unicast, esto es debido a que el suscriptor ya está asociado a tal dato o parámetro de frecuencia, de este modo ya se formó una comunicación de un único emisor a un único receptor en el análisis de este paquete. En la siguiente prueba se podrá ver como esto cambia cuando el dato es recogido por un elemento que no es ni publicador ni receptor asociado a este parámetro.

5.6.2 Paquetes recogidos desde una Pc sin ser publicador ni suscriptor.

The screenshot displays the Wireshark interface for a capture named 'Escucha en transmision DDS (2Pc's).pcap'. The packet list pane shows several RTPS2 packets. The selected packet (No. 68) is detailed in the packet details pane, showing it is an RTPS2 packet (Real-Time Publish-Subscribe wire Protocol 2.x) with source 192.168.5.102 and destination 239.255.0.1. The packet bytes pane shows the hex and ASCII representation of the packet, with the ASCII part containing the text '..b.(. RTI Data Distribution Service Spy....'.

Figura 31: Paquete 100 de transmisión de frecuencia de 60Hz por DDS visto desde una Pc cualquiera. Fuente: Propia.

En la figura 31 se puede apreciar que cuando el paquete es recogido por una Pc externa a DDS, existe una dirección IP de fuente que en este caso es el publicador del DDS pero no existe una dirección conocida de destino, esto es porque los paquetes que no tienen un suscriptor asociado dan aviso de que tales datos están presentes en la red y que algún suscriptor asociado lo pudiera

tomar sin ningún problema. Por otro lado, si en esta Pc existiera un suscriptor con el nombre de tópico correspondiente lo podría tomar o recibir y el paquete tendría una dirección destino correspondiente y conocida. Otra situación que se puede observar que el paquete tiene 810 bytes de igual manera que en la prueba anterior y sigue siendo UDP y RTPS2 que es el protocolo de DDS. Por último la figura muestra que el paquete no tiene dato alguno sino que es solo información de que este dato con ese tópico están presentes en la red de comunicación.

Discusión:

El análisis de estos paquetes es interesante debido a que se puede ver con claridad que DDS no inunda la red con todos los datos que están viajando por la red para que en algún momento un suscriptor los recoja, sino que envía avisos de que tal dato con tal tópico están presentes y que si hay algún suscriptor que la quiere tomar que se asocie o se suscriba para recibir el dato en cuestión. También en la figura se puede ver que el paquete es ahora multicast el cual es repartido a los miembros de la red DDS para dar a conocer su presencia.

5.7 Comparación del estándar DDS (UDP) Vs. Aplicaciones (TCP).

Esta última prueba se trata de comparar las transmisiones de frecuencia por DDS que utiliza en su protocolo de transporte UDP y por otra aplicación que usa TCP. En principio para esto se deberá redactar algunas líneas de código en el programa freq.c para que tal programa sea capaz de recoger los datos muestreados de la señal eléctrica, calcular el resultado de frecuencia y enviar estos datos con la aplicación TCP. A continuación se muestran los resultados obtenidos del conteo, tamaño y velocidad de los paquetes tanto de DDS como de la aplicación TCP.

DDS				Aplicación TCP			
Frecuencia (Hz)	Paquetes/seg	Tamaño de un paquete (Bytes)	Bytes/seg	Frecuencia (Hz)	Paquetes/seg	Tamaño de un paquete (Bytes)	Bytes/seg
40	40	118	4720	40	39	134	5226
45	45	118	5310	45	44	134	5896
50	50	118	5900	50	49	134	6566
55	54	118	6372	55	54	134	7236
60	58	118	6844	60	60	134	8040
65	64	118	7552	65	65	134	8710
70	69	118	8142	70	70	134	9380
75	74	118	8732	75	76	134	10184

Tabla 4: Comparación de resultados de medida entre DDS y aplicaciones TCP.

La tabla 4 muestra como DDS y a la aplicación TCP actúan en la red de comunicación y se puede observar también que mientras sube la frecuencia sube también la cantidad de paquetes enviados por segundo, esto es así ya que el cálculo de la frecuencia está ligado directamente al tiempo real

transcurrido. Estos se puede afirmar con los datos recogidos y puestos en la tabla en la que indican que el valor de frecuencia en Hz es muy parecido al valor de paquetes por segundo.

Discusión:

Luego de realizar las pruebas se obtuvieron los datos que aparecen en la tabla 4, en estos se puede ver que los resultados de la una y de la otra transmisión son distintos. El paquete enviado por DDS es más pequeño que el paquete enviado por la aplicación TCP. Esto era de esperarse puesto que un paquete UDP tiene menos datos de señalización que uno TCP lo que conlleva a tener más Bytes transmitidos en un tiempo dado. La limitante para alcanzar más velocidad en la red es el tiempo que se ocupa para el cálculo de la frecuencia en el SBC. Además en este caso si seguiríamos transmitido paquetes hasta saturar la red, se podría enviar mayor cantidad de paquetes por DDS que por la aplicación TCP ya que el tamaño del paquete es más pequeño. Si tenemos una red de 10Mbps y un paquete DDS en 60Hz tiene 118Bytes se pueden enviar hasta 10593 paquetes por segundo, pero si tenemos la misma red con paquetes que corresponden a la aplicación TCP en 60Hz con un tamaño de 134Bytes se pueden enviar hasta 9328 paquetes por segundo.

6 CONCLUSIONES Y TRABAJO FUTURO

6.1 Conclusiones

Con la investigación de DDS se pudo averiguar que muchas industrias y empresas incorporan este estándar en su sistema de comunicación para satisfacer una parte o toda su estructura de intercambio de datos sean estos de control, monitoreo o administración. Lo anteriormente mencionado es una realidad gracias a que este estándar es confiable, tiene varios años en el mercado y es capaz de hacer frente a las necesidades de cada uno de sus clientes que requieren una comunicación oportuna y eficaz, siendo útil para un gran conjunto de sistemas hoy en día. DDS además posee ventajas en comparación con otros estándares de comunicación para micro-redes ya que trabaja con el modelo publicador-suscriptor, cuenta con el manejo de varias calidades de servicios, tiene una arquitectura flexible y adaptable, auto-descubrimiento y un gran número de parámetros configurables lo que proporciona a los desarrolladores un gran control del sistema. También se investigó que EIC61850 es un sistema de estandarización usado en el diseño y automatización de subestaciones eléctricas el cual permite involucrar elementos de distintos fabricantes siempre que cumplan con tal estándar. Esto es realmente bueno ya que DDS puede trabajar junto con IEC para hacer aun más eficiente el uso de este estándar de comunicación en subestaciones y micro-redes.

En la incorporación del SBC (Raspberry Pi) como elemento de medida, se pudo observar que este es de mucha ayuda cuando se trata de realizar tareas dentro de una micro-red, ya que es un componente capaz de guardar registros, procesar información, tomar decisiones y algunas otras acciones que favorecen al control y gestión de la micro-red. Por ejemplo un suscriptor puede recoger los datos de frecuencia de la red de comunicación y funcionar a la vez como un publicador tomando algún tipo de acción cuando esta medida varíe, además pudiera adicionalmente enviar estos datos a la red de comunicación DDS para que otro elemento tome esta información y registre el evento. También es preciso recordar que es posible trabajar con este mini computador para dejar de utilizar un sistema de comunicación y administración totalmente centralizado y pasar a usar uno semi-centralizado o quizá distribuido, ya que el Raspberry Pi es muy funcional y capaz de realizar varias tareas a la vez y con precisión.

En la red de comunicación para la prueba de concepto se interconectaron los computadores y computadores de placa (SBC) que constituyen los elementos de la red DDS en la cual se logro la transmisión efectiva de la frecuencia eléctrica por el estándar DDS a través de todos sus elementos publicadores y suscriptores con esto se verificó el buen funcionamiento del estándar. Adicionalmente con la recolección y análisis de paquetes en la red y el manejo de tópicos se vió la verdadera potencia de DDS en la transmisión de datos, uso de la red y velocidad. Se pudo corroborar con los resultados obtenidos que los paquetes que maneja DDS son de menor tamaño comparados con otros, pueden ser paquetes unicast y multicast según sea el caso y pueden tener calidades de servicio.

En el análisis de paquetes DDS también se pudo observar la ausencia de suficiente seguridad, con esto se pretende informar que la versión del estándar con el que se trabajo, no posee un sistema

de protección de datos o un mecanismo de encriptación. Hace falta un sistema de seguridad que proteja los datos que circulan por la red pero tal medida traerá un pequeño desbalance con la velocidad. Por tal razón es preciso tener un equilibrio entre estos dos aspectos que son la seguridad y la velocidad.

6.2 Trabajo futuro

Este proyecto de tesis surgió con la idea de poder implementar el estándar de comunicación DDS en el control y gestión de una micro-red real por lo que, en primer lugar, se podría trabajar a futuro en la elaboración de otros sistemas para la medición y pruebas de parámetros de operación de las micro-redes como son voltajes, corrientes, estado de carga de baterías y velocidad de turbinas eólicas por nombrar algunos. Estos sistemas se pueden desarrollar con la implementación de computadores de placa que trabajen con frecuencias de muestreo más altas para desarrollar además aplicaciones de mayor velocidad de reacción como fallas del sistema. Luego se podría continuar con el desarrollo de otras configuraciones de programación y pruebas del estándar DDS en temas como calidad de servicio, nodos y dominios entre otros con el propósito de conocer más sobre el estándar

Adicionalmente se puede continuar con la investigación para conocer como DDS trabaja con la norma IEC61850, esto conlleva la implementación minuciosa de tópicos con los que cuenta DDS y las especificaciones de IEC. Posteriormente se podría realizar pruebas con instrumentos reales que cumplan con la norma para finalmente tratar de construir un software de control para la micro-red.

7 BIBLIOGRAFÍA

- [1] Z. JACOBSON, A. DELUCCHI, A.F. BAUER. 2015. "100% Clean and Renewable Wind, Water, and Sunlight (WWS)". [en línea]. Energy & Environmental Science. Stanford Estados Unidos. < <http://web.stanford.edu/group/efmh/jacobson/Articles/I/USStatesWWS.pdf> >. [consulta: 09 agosto 2015].
- [2] HIGH-LEVEL PANEL ON GLOBAL SUSTAINABILITY. 2012. "Resilient People, Resilient Planet: A future worth choosing". [en línea]. New York Estados Unidos . < http://en.unesco.org/system/files/GSP_Report_web_final.pdf >. [consulta: 21 Enero 2015].
- [3] UN-HABITAT. 2011. "Cities and Climate Change", [en línea]. Global Report on Human Settlements. Washington DC Estados Unidos. < <http://unhabitat.org/books/cities-and-climate-change-global-report-on-human-settlements-2011/> > [consulta: 21 Enero 2015].
- [4] R. LASSETER. 2011. "Smart Distribution: Coupled Microgrids". IEEE. USA. 99(6) 1 – 4
- [5] CIGRÉ. 2013. "Microgrids". cigré C6. [en línea]. < <http://c6.cigre.org/WG-Area/WG-C6.22-Microgrids> >
- [6] N. HATZIARGYRIOU. 2007. "Microgrids". Power and Energy Magazine. Atenas Grecia. IEEE.vol. 5, pp.78-94, 2007.
- [7] C. COLSON, M. NEHRIR. 2009. "A review of challenges to real-time power management of Microgrids". Power & Energy General Meeting. IEEE. USA. 09 1 - 8.
- [8] F. KATIRAEI, R. IRAVANI, N. HATZIARGYRIOU. 2008. "A. Dimeas, Microgrids management".Power and Energy Magazine. IEEE. Varennes Canada. 6(3). 54–65.
- [9] M. DICORATO, A. CAGNANO. 2014. "The Application of Communication Architectures for the Management of an Experimental Microgrid". *CIREN Workshop*. [en línea]. Roma Italia. < http://www.cired.net/publications/workshop2014/papers/CIREN2014WS_0329_final.pdf >. [consulta: 10 abril 2015].
- [10] L. SIOW, P. SO, H. GOOI, F. LUO, C. GAJANAYAKE, Q. YO. 2009. "Wi-Fi based server in Microgrid energy management system". Region 10 Conference, TENCON. IEEE. Singapore. 1 – 5.

- [11] S. THALE, V. AGARWAL. 2011. "Design and implementation of communication and control architecture for solar PV based Microgrid supported by PEM fuel cell based auxiliary source". Photovoltaic Specialists Conference (PVSC). IEEE. Mumbai, India.
- [12] ZHU, YIXIN, F. ZHUO, AND L. XIONG. 2012. "Communication platform for energy management system in a master-slave control structure Microgrid". Power Electronics and Motion Control Conference (IPEMC). IEEE. Harbin China. 1 – 4.
- [13] Y. BI, L. JIANG, MEMBER, X.J. WANG AND L.Z. CUI. 2013. "Mapping of IEC 61850 to Data Distribute Service for Digital Substation Communication", Power and Energy Society General Meeting (PES), IEEE. Jinan, China. 1-5.
- [14] STANCIULESCU, G. 2012. "Communication technologies for BCIT Smart Microgrid". Innovative Smart Grid Technologies (ISGT). IEEE. Canada. 1 – 7.
- [15] D. PURUSOTHAMAN, RAJESH, VIJAYARAGHAVAN AND BAJAJ. 2014. "Design of Arduino-based Communication Agent for Rural Indian Microgrids". Innovative Smart Grid Technologies. IEEE. Asia.
- [16] BONFIGLIO, BARILLARI, DELFINO, PAMPARARO, ROSSI, INVERNIZZI, DENEGRI AND BRACCO. 2013. "The smart microgrid pilot project of the University of Genoa: Power and communication architectures." AEIT Annual Conference. IEEE. Genoa, Italy. 1 – 6.
- [17] H.SCHULZRINNE, A. RAO, R. LANPHIER. 1998. "Real Time Streaming Protocol (RTPS)". RFC 2326. [en linea]. < <https://www.ietf.org/rfc/rfc2326.txt> > [consulta: 19 Septiembre 2015].
- [18] J. MARTINEZ. 2012. "DDS en SCADA y Utilities.Mp4". NEXTEL Aerospace Defence & Security S.L. [en linea]. Madrid España. < <https://www.youtube.com/watch?v=ibwwKSPmcZM> >. [consulta: 5 May 2015].
- [19] OMG.ORG. 2015. " Data Distribution Service (DDS)". Version 1.4. [en linea]. < <http://www.omg.org/spec/DDS/1.4> >. [consulta: 5 May 2015].
- [20] OMG.ORG. 2014. " The Real-time Publish-Subscribe Protocol (RTPS) DDS Interoperability Wire Protocol Specification". Version 2.2. [en linea]. < <http://www.omg.org/spec/DDS-RTPS/2.2> >. [consulta: 5 May 2015].
- [21] G. PRADO-CASTELLOTE. 2005. "Data Distribution Service Tutorial". Real-Time Innovations, Inc. [en linea]. California Estados Unidos. < https://www.rti.com/docs/DDS_Enabling_Global_Data.pdf >. [consulta: 3 May 2015].

- [22] I. RODRÍGUEZ LÓPEZ. "Desarrollo de Software Distribuido de Tiempo Real basado en DDS". Proyecto de Ingeniería Técnica de Telecomunicación. Madrid España. Universidad Carlos III de Madrid. 159p
- [23] RTI.COM. 2015. " Downloads for RTI Connex DDS Professional". Connex DDS software. [en línea]. < <http://www.rti.com/downloads/connex-files.html> > [consulta: 15 Nov 2014].
- [24] A. CORSARO. 2010. "The DDS Tutorial Part 1" Primstech Power Netcentriciy. [en línea]. < http://www.laas.fr/files/SLides-A_Corsaro.pdf > [consulta: 8 Febrero 2015].
- [25] R. QUINTANILLA R., J. YARZA. 2010. "Nuevas Exigencias y Aplicaciones de Comunicación para la Protección de Micro-redes". Revista CIER N° 57. [en línea]. Brasil. < [http://sg.cier.org.uy/Publicaciones/Revista.nsf/0a293b20eacdf8a903257133003ea67d/38d5383c47c86f238325782c004a2df5/\\$FILE/07_NuevasExigencias.pdf](http://sg.cier.org.uy/Publicaciones/Revista.nsf/0a293b20eacdf8a903257133003ea67d/38d5383c47c86f238325782c004a2df5/$FILE/07_NuevasExigencias.pdf) >. [consulta: 22 enero 2015].
- [26] E. INGA. 2012 "Redes de Comunicación en Smart Grid". INGENIUS Revisa de Ciencia y Tecnología. 33 – 55.
- [27] COOKING-HACKS. "Raspberry Pi to Arduino Shields Connection Bridge". [on line]. < <https://www.cooking-hacks.com/documentation/tutorials/raspberry-pi-to-arduino-shields-connection-bridge/> > [consulta: 09 May 2015].

ANEXO A

TUTORIAL PARA LA INSTALACION DE DDS DE RTI EN EL COMPUTADOR.

Contenido

TUTORIAL PARA LA INSTALACION DE DDS DE RTI EN EL COMPUTADOR.....	67
1.- DDS.....	67
2.- Sistema Operativo CENTOS.....	67
3.- Instalación y Configuración del Sistema Operativo Centos 6.0.....	68
4.- Instalación de Java y GCC.....	71
5.- Descarga e instalación de DDS de RTI.....	72
6.- Configuración de DDS de RTI.....	74
7.- Ejecutar un ejemplo para probar DDS de RTI.....	77

1.- DDS.

Data Distribution Service (DDS) es la especificación del middleware para estandarizar el modelo de programación de data-centric publish-subscribe para sistemas distribuidos. Simplifica la compleja programación de red implementando el modelo de publicación-suscripción para enviar y recibir datos, eventos, y comandos entre los nodos. Los nodos que producen información (publicadores) crean tópicos (por ejemplo, temperatura, localización, presión, etc.) y publican muestras de estos tópicos. RTI Data Distribution Service entrega estas muestras a todos los suscriptores que declaren su interés en el tópico en cuestión. DDS gestiona todas las fases de la transferencia: direccionamiento de los mensajes, la serialización y deserialización a formato estándar (así, los suscriptores pueden estar en diferentes plataformas a la del publicador), entrega, control de ancho de banda, reintentos, etc. Cualquier nodo puede ser un publicador, suscriptor, o ambas cosas simultáneamente.

En el presente tutorial se tratarán diferentes puntos acerca de la instalación y funcionamiento de DDS de RTI en un sistema operativo CENTOS.

2.- Sistema Operativo CENTOS.

El Community ENTERprise Operating System (CENTOS) es un sistema operativo de Linux de fuente abierta, basado en la distribución Red Hat Enterprise Linux. Destinado a ser un sistema de programa de "clase empresarial" gratuito, Centos es robusto, estable y fácil de instalar y utilizar. Luce y se opera de forma similar al RHEL (Red Hat Enterprise Linux).

3.- Instalación y Configuración del Sistema Operativo Centos 6.0.

Se recomienda la instalación de este sistema operativo Centos para el buen funcionamiento de DDS de RTI ya que ambos trabajan en base a una distribución Red Hat.

Para la instalación de Centos primero se deberá descargar la versión más adecuada para su Pc o maquina virtual, sea esta de 64 o 32 Bytes, para este tutorial se dispondrá de una Pc antigua de 32 bits y 512MB de Ram por lo que se descargara una versión comprimida de Centos.

Los pasos a seguir para la instalación son los siguientes:

Descargar Centos

Cualquier versión del sistema operativo se podrá descargar desde el siguiente enlace:

ftp://ftp.inf.utfsm.cl/pub/Linux/CentOS/6.5/isos/

El sistema operativo que se utilizara en este tutorial está en la dirección:

ftp://ftp.inf.utfsm.cl/pub/Linux/CentOS/6.5/isos/i386/

que hace referencia específicamente al archivo CentOS-6.5-i386-minimal.iso

Esta versión tiene un tamaño de 324Mb y se debe grabar en un CD para instalarlo en la Pc o maquina virtual.

Instalar Centos

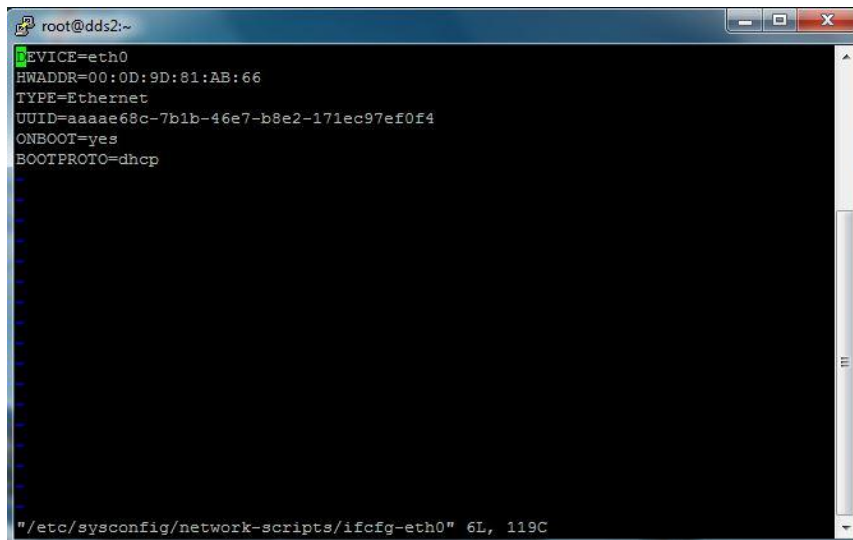
Para la instalación es necesario configurar la máquina para que arranque desde el CD y se proceda con la instalación. En la maquina virtual es necesario indicar la imagen que se descargo anteriormente y seguir los pasos.

Una vez instalado la versión comprimida de Centos es necesario descargar el paquete que contiene la versión grafica de Centos, para esto hay que configurar en modo consola la tarjeta de red de modo que pueda obtener una dirección IP y acceder a internet.

Para obtener una dirección IP automáticamente es necesario digitar los siguientes comandos:

vi /etc/sysconfig/network-scripts/ifcfg-eth0

El comando permite entrar a configurar el archivo ifcfg-eth0 el cual debe quedar como se indica a continuación



```
root@dds2:~  
DEVICE=eth0  
HWADDR=00:0D:9D:81:AB:66  
TYPE=Ethernet  
UUID=aaaae68c-7b1b-46e7-b8e2-171ec97ef0f4  
ONBOOT=yes  
BOOTPROTO=dhcp  
...  
"/etc/sysconfig/network-scripts/ifcfg-eth0" 6L, 119C
```

Archivo ifcfg-eth0

Para salir del archivo guardando los cambio

ESC:x (La tecla ESC mas :x)

Para salir del archivo sin guardar los cambios

ESC:q! (La tecla ESC mas :q!)

Para finalizar se debe reiniciar la tarjeta de red con el comando:

service network restart

Para probar que efectivamente estamos conectados a internet se puede dar un ping a www.google.com.

Si el pin no es correcto se recomienda reiniciar la Pc.

Es recomendable descargar el editor de textos nano con:

yum install nano

Luego descargar e instalar el paquete grafico de Centos con el siguiente comando:

yum -y groupinstall "Desktop" "Desktop Platform" "X Window System" "Fonts"

```
root@sys01svr01:~
Transaction Summary
-----
Install      478 Package(s)

Total download size: 286 M
Installed size: 888 M
Downloading Packages:
(1/478): ConsoleKit-0.4.1-3.el6.x86_64.rpm           | 82 kB    00:00
(2/478): ConsoleKit-libs-0.4.1-3.el6.x86_64.rpm      | 17 kB    00:00
(3/478): ConsoleKit-x11-0.4.1-3.el6.x86_64.rpm      | 20 kB    00:00
(4/478): DeviceKit-power-014-3.el6.x86_64.rpm       | 91 kB    00:00
(5/478): GConf2-2.28.0-6.el6.x86_64.rpm            | 964 kB   00:00
(6/478): GConf2-gtk-2.28.0-6.el6.x86_64.rpm         | 23 kB    00:00
(7/478): ModemManager-0.4.0-5.git20100628.el6.x86_64.rpm | 177 kB   00:00
(8/478): NetworkManager-0.8.1-66.el6.x86_64.rpm     | 1.1 MB   00:00
(9/478): NetworkManager-glib-0.8.1-66.el6.x86_64.rpm | 230 kB   00:00
(10/478): NetworkManager-gnome-0.8.1-66.el6.x86_64.rpm | 463 kB   00:00
(11/478): ORBit2-2.14.17-3.2.el6_3.x86_64.rpm       | 168 kB   00:00
(12/478): PackageKit-0.5.8-21.el6.x86_64.rpm        | 526 kB   00:00
(13/478): PackageKit-device-rebind-0.5.8-21.el6.x86_64.r | 95 kB    00:00
(14/478): PackageKit-glib-0.5.8-21.el6.x86_64.rpm   | 221 kB   00:00
(15/478): PackageKit-gtk-module-0.5.8-21.el6.x86_64.rpm | 95 kB    00:00
(16/478): PackageKit-yum-0.5.8-21.el6.x86_64.rpm    | 156 kB   00:00
```

Instalación de Centos modo grafico

Al finalizar la instalación se continuará con la modificación el siguiente archivo:

`nano /etc/inittab`

Nos ubicamos en la línea `id:3:inittdefault:` y se cambia el numero 3 (por defecto) por el 5.

y deberá quedar de la siguiente manera:

```
root@dds2:~
GNU nano 2.0.9      File: /etc/inittab
#
# Terminal gettys are handled by /etc/init/tty.conf and /etc/init/serial.conf,
# with configuration in /etc/sysconfig/init.
#
# For information on how to write upstart event handlers, or how
# upstart works, see init(5), init(8), and initctl(8).
#
# Default runlevel. The runlevels used are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id:5:inittdefault:
^G Get Help  ^C WriteOut  ^R Read File ^Y Prev Page  ^X Cut Text   ^C Cur Pos
^X Exit      ^O Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Configuración de arranque del sistema

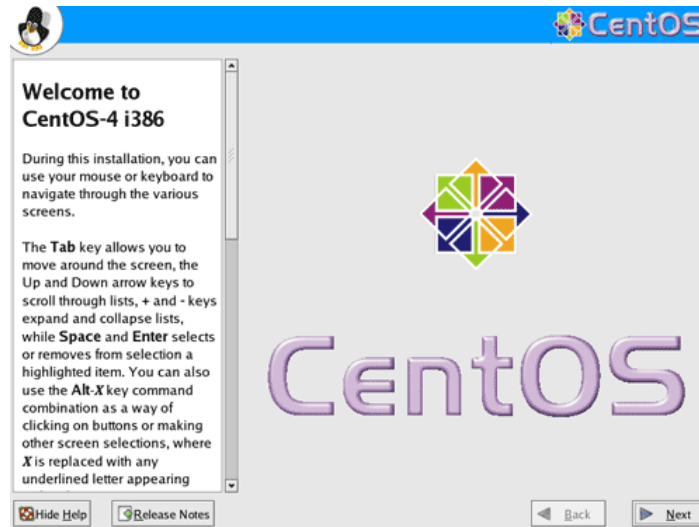
Para salir de este archivo presionamos CTRL + x y sin olvidar grabar los cambios.

Digitamos

`init 6`

Esto hará que la computadora se reinicie.

Cuando la computadora se haya reiniciado mostrara en la pantalla la imagen que se muestra a continuación, a partir de este punto la instalación se puede realizar fácilmente siguiendo los pasos señalados.



Instalación de Centos modo grafico.

4.- Instalación de Java y GCC.

Antes de instalar directamente DDS de RTI es necesario verificar que el equipo tenga instalado Java y el compilador GCC. Estos dos programas son importantes ya que DDS de RTI los ocupa para su correcto funcionamiento. Las versiones de estos programas varía de acuerdo al Centos instalado. Para este caso como tenemos instalado Centos 6.0 se requiere la versión mínima Java (JDK) 1.7 y la versión mínima gcc 4.4.5. Estos datos los podemos extraer también de la propia página de RTI.

Para instalar Java en la Pc se deberá digitar el siguiente código:

```
yum install java-1.7.0-openjdk java-1.7.0-openjdk-devel
```

El código para instalar gcc es:

```
yum install gcc
```

Luego de la instalación de estos dos componentes se procederá a verificar que las versiones sean las correctas, para esto digitamos los códigos que muestra la siguiente figura.

```
root@dds2:~  
[root@dds2 ~]# java -version  
java version "1.7.0_65"  
OpenJDK Runtime Environment (rhel-2.5.1.2.el6_5-1386 u65-b17)  
OpenJDK Client VM (build 24.65-b04, mixed mode, sharing)  
[root@dds2 ~]# gcc --version  
gcc (GCC) 4.4.7 20120313 (Red Hat 4.4.7-4)  
Copyright (C) 2010 Free Software Foundation, Inc.  
This is free software; see the source for copying conditions. There is NO  
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  
[root@dds2 ~]#
```

Versiones de Java y Gcc.

Luego de esto nuestro equipo estará listo para la descarga e instalación de DDS de RTI.

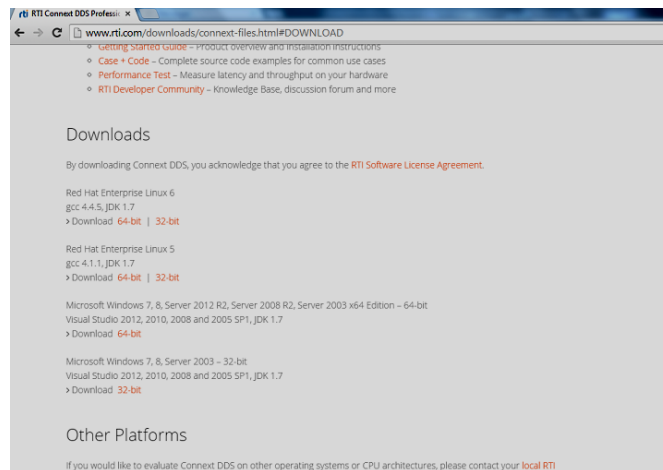
5.- Descarga e instalación de DDS de RTI.

El software y ejemplos de DDS de RTI están disponibles en la sitio web de RTI que es www.rti.com/

en donde es necesario registrar los datos y correo de la persona para realizar la respectiva descarga. RTI proporciona una licencia de prueba 30 días para realizar pruebas de DDS. La universidad dispone de una licencia que dura hasta junio del 2015 por lo que se utilizará esta para ver el funcionamiento de DDS de RTI.

La licencia tiene el nombre de: *rti_license.dat*

El sitio web de RTI para la descarga del programa de aplicación se muestra en la siguiente figura.



Sitio web de descarga de DDS de RTI.

La dirección completa de la página mostrada en la anterior figura es:

<http://www.rti.com/downloads/connext-files.html#DOWNLOAD>

En esta página encontraremos algunas versiones de DDS de RTI compatibles tanto para sistemas Red Hat como para sistemas Windows.

Para encontrar otras versiones para otros sistemas operativos y dispositivos como por ejemplo: RTI Connex DDS Professional, RTI Connex DDS Linux Live ISO (incluye RTI Connex DDS Professional), RTI Connex DDS Libraries for Raspberry Pi, etc es necesario ir al enlace de RTI Community que se encuentra en la siguiente dirección:

<http://community.rti.com/downloads>

Descarga de DDS de RTI para Red Hat (Centos).

En este tutorial se descargará la siguiente versión DDS de RTI:

Red Hat Enterprise Linux 6 gcc 4.4.5, JDK 1.7 | 32-bit

ya que esta versión es compatible con los programas GCC y Java anteriormente instalados.

El programa se descargará con el siguiente nombre:

RTI_Connext_DDS_Professional-5.1.0-RHEL6_32_lic.sh

Una vez descargado el programa se procederá a realizar la respectiva instalación, para antes debemos asegurarnos que el archivo .sh este bajo el directorio /opt, de no ser así se tendrá que mover.

Instalación de DDS de RTI en Centos.

Una vez que el archivo se encuentre en /opt de la siguiente manera

/opt/RTI_Connext_DDS_Professional-5.1.0-RHEL6_32_lic.sh

se podrá realizar la instalación del programa digitando en siguiente comando:

[root@dds3 opt]# ./RTI_Connext_DDS_Professional-5.1.0-RHEL6_32_lic.sh

Se iniciara la instalación del programa y aparecerá un mensaje de la siguiente indicación:

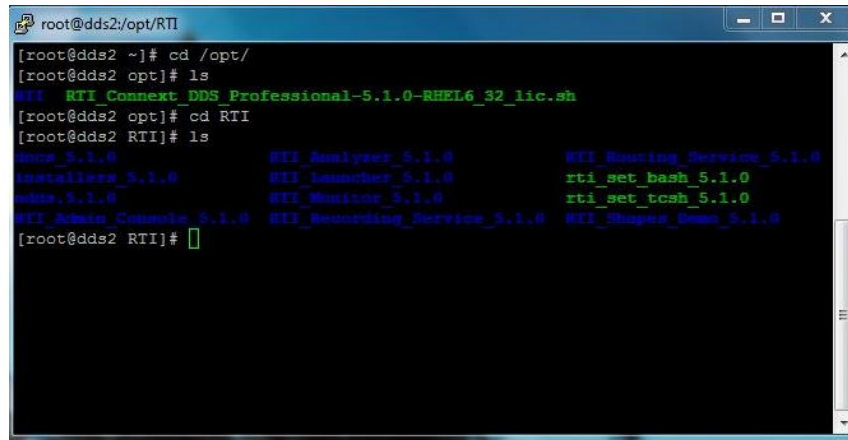
Specify a location for the installation (the directory "RTI" will be created in the location you specify): [/root]

Luego que aparezca este mensaje digitar

/opt (y luego presionar enter para que continúe).

El mensaje anterior es para señalar la ruta en la que se quiere instalar DDS de RTI.

Al finalizar la instalación el directorio /opt quedara como indica la siguiente figura.



```
root@dds2/opt/RTI
[root@dds2 ~]# cd /opt/
[root@dds2 opt]# ls
RTI RTI_Connect_DDS_Professional-5.1.0-RHEL6_32_lic.sh
[root@dds2 opt]# cd RTI
[root@dds2 RTI]# ls
docs_5.1.0 RTI_Analyzer_5.1.0 RTI_Routing_Service_5.1.0
installers_5.1.0 RTI_Launcher_5.1.0 rti_set_bash_5.1.0
ndds_5.1.0 RTI_Monitor_5.1.0 rti_set_tcsh_5.1.0
RTI_Admin_Console_5.1.0 RTI_Recording_Service_5.1.0 RTI_Shapes_Demo_5.1.0
[root@dds2 RTI]#
```

Directorio /opt y su contenido.

Como se puede ver, luego de la instalación del programa se genero una carpeta RTI con 12 archivos en su interior con los siguientes nombres:

- docs_5.1.0
- RTI_Analyzer_5.1.0
- RTI_Routing_Service_5.1.0
- installers_5.1.0
- RTI_Launcher_5.1.0
- rti_set_bash_5.1.0
- ndds.5.1.0
- RTI_Monitor_5.1.0
- rti_set_tcsh_5.1.0
- RTI_Admin_Console_5.1.0
- RTI_Recording_Service_5.1.0
- RTI_Shapes_Demo_5.1.0
- rti_license.dat (Este archivo deberá ser removido posteriormente por lo que no se muestra en la imagen).

Es muy importante asegurarse que se encuentren instalados todos los archivos que se presentaron anteriormente para garantizar una adecuada descarga e instalación de DDS de RTI.

6.- Configuración de DDS de RTI.

Una vez instalado DDS es necesario realizar ciertas configuraciones antes de probar los ejemplos de DDS, para empezar esto se debe dirigir a la siguiente ruta:

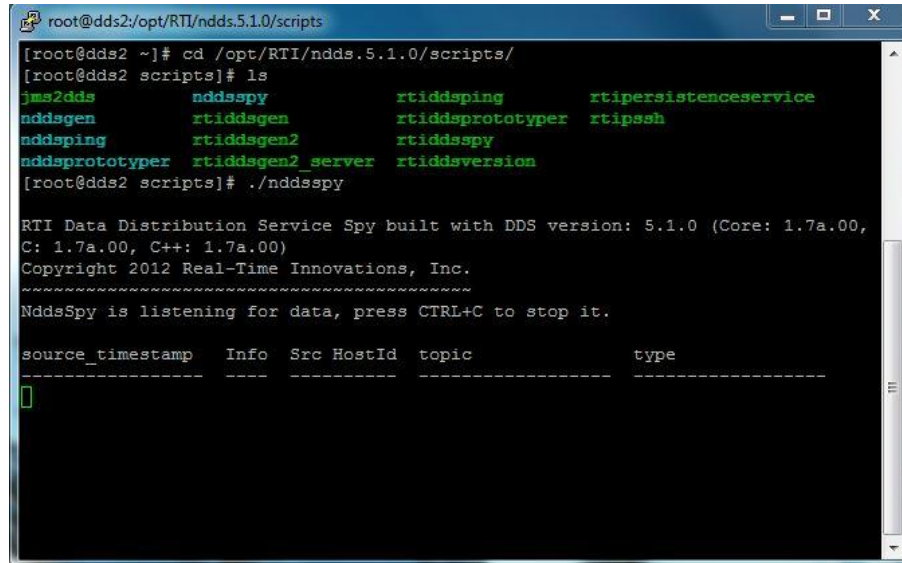
```
cd /opt/RTI/ndds.5.1.0/scripts/
```

para luego ejecutar el siguiente comando:

./nddsspy

Al correr el programa `nddsspy` se activa el DDS de RTI para poder correr los ejemplos que trae incluido el software y debe mostrar la información como muestra la siguiente figura.

Nota: antes de ejecutar el programa `nddsspy` es necesario que la Pc este conectada a una red para en esta realizar luego las pruebas y que conecte a internet.



```
root@dds2/opt/RTI/ndds.5.1.0/scripts
[root@dds2 ~]# cd /opt/RTI/ndds.5.1.0/scripts/
[root@dds2 scripts]# ls
jms2dds          nddsspy          rtiddsping       rtipersistenceservice
nddsngen        rtiddngen        rtiddsprototyper rtipssh
nddssping       rtiddngen2       rtiddsspy
nddsprototyper  rtiddngen2_server rtiddsversion
[root@dds2 scripts]# ./nddsspy

RTI Data Distribution Service Spy built with DDS version: 5.1.0 (Core: 1.7a.00,
C: 1.7a.00, C++: 1.7a.00)
Copyright 2012 Real-Time Innovations, Inc.
~~~~~
NddsSpy is listening for data, press CTRL+C to stop it.

source_timestamp  Info  Src HostId  topic  type
-----
█
```

Ejecución del programa `nddsspy`

La herramienta `nddsspy` que incluye DDS de RTI permite monitorizar las los DataWriters, DataReaders y las muestras de datos que aparecen en una red. De esta forma se controla la correcta creación y destrucción de entidades remotas, así como de los datos que se publican.

El siguiente paso de la configuración es realizar la declaración de la variable de entorno en el sistema operativo Centos. Una variable del entorno es un valor dinámico cargado en la memoria, que puede ser utilizado por varios procesos que funcionan simultáneamente.

Para la declaración de la variable de entorno entramos el archivo `.bashrc` con el editor de textos `nano` de la siguiente manera:

```
[root@dds2 ~]# nano .bashrc
```

En este caso como estamos trabajando como `root` el archivo `.bashrc` en donde se declara la variable de entorno está en la siguiente ubicación:

/root

En este archivo `.bashrc` se digitara las siguientes instrucciones:

```
export NDDSHOME=/opt/RTI/ndds.5.1.0
```

```
export PATH=$NDDSHOME/scripts:/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.65/bin:$PATH
```

```
export LD_LIBRARY_PATH=$NDDSHOME/lib/i86Linux2.6gcc4.4.5jdk:$LD_LIBRARY_PATH
```

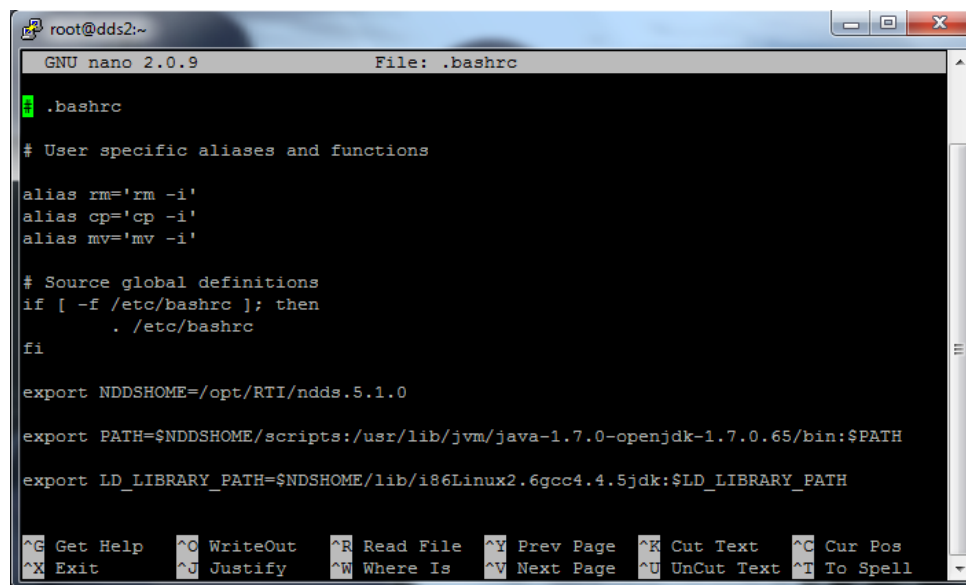
NDDSHOME es una variable de entorno que direcciona a la ruta de ndds.5.1.0.

Es necesario revisar cuidadosamente estos códigos ya que la primera línea debe apuntar a la ruta en donde se encuentre la carpeta ndds.5.1.0. La segunda línea hace referencia a los directorios del programa jdk (Java) específicamente apuntara a la carpeta /bin del grupo de archivos del programa Java 1.7 instalado anteriormente. La tercera línea está relacionada con la archivo bin que está dentro de la carpeta ndds.5.1.0.

Es recomendable comprobar estas rutas que estén en las direcciones correcta antes de guardar y salir del editor de textos nano.

Luego de realizar la digitación de este archivo para definir la variable es necesario reiniciar la PC.

La siguiente figura muestra el contenido del archivo .bashrc

A screenshot of a terminal window showing the GNU nano 2.0.9 editor editing the .bashrc file. The terminal title is 'root@dds2:~'. The editor content includes: '# User specific aliases and functions', 'alias rm='rm -i'', 'alias cp='cp -i'', 'alias mv='mv -i'', '# Source global definitions', 'if [-f /etc/bashrc]; then', ' . /etc/bashrc', 'fi', 'export NDDSHOME=/opt/RTI/ndds.5.1.0', 'export PATH=\$NDDSHOME/scripts:/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.65/bin:\$PATH', 'export LD_LIBRARY_PATH=\$NDDSHOME/lib/i86Linux2.6gcc4.4.5jdk:\$LD_LIBRARY_PATH'. The bottom of the window shows nano editor shortcuts: '^G Get Help', '^O WriteOut', '^R Read File', '^Y Prev Page', '^K Cut Text', '^C Cur Pos', '^X Exit', '^J Justify', '^W Where Is', '^V Next Page', '^U UnCut Text', '^T To Spell.

Modelo de archivo .bashrc luego de las configuraciones.

Para comprobar si la variable NDDSHOME está funcionando correctamente digitamos:

```
cd $NDDSHOME
```

El comando debe ubicarnos en la ruta:

```
/opt/RTI/ndds5.1.0
```

7.- Ejecutar un ejemplo para probar DDS de RTI.

DDS de RTI contiene ejemplos para poder probar la teoría de transmisión de archivos en los cuales se puede apreciar fácilmente el concepto de suscriptor y publicador.

El ejemplo que trae RTI que se probara consiste en tener un publicador que envíe datos cualesquiera a la red para que un determinado suscriptor los reciba teniendo en cuenta que este debe estar configurado o suscrito para recibir los datos del publicador.

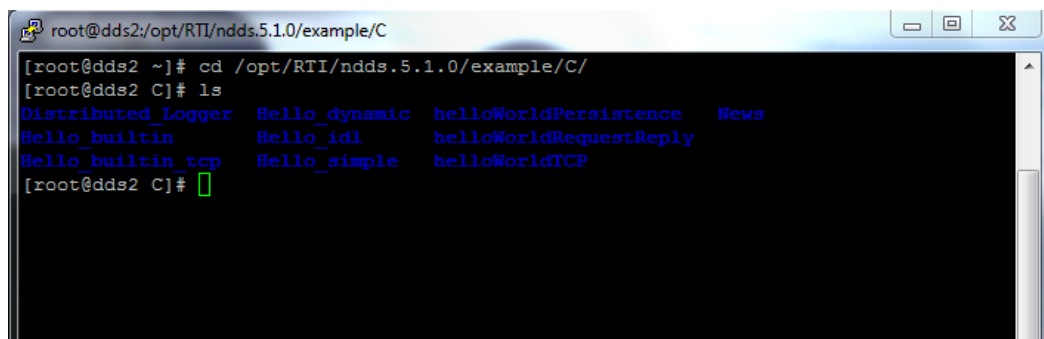
El suscriptor puede estar en la misma Pc o en otra en la que este instalado de igual manera DDS de RTI y dentro de la misma red LAN.

Luego de probar teniendo en la misma computadora un publicador y un suscriptor enviando y recibiendo datos se podrá ampliar la red a mas PC's o incluso ordenadores de placa reducida.

Uno de los ejemplos de aplicación que trae DDS de RTI es Hello_simple.

Pasos para probar el ejemplo Hello_simple:

1. Para comenzar con las pruebas de este ejemplo es necesario ir a revisar el directorio:
[root@dds2 ~]# cd \$NDDSHOME/examples/C/
2. En este se encuentra el archivo Hello_simple el cual se indica en la siguiente figura:



```
root@dds2:/opt/RTI/ndds.5.1.0/example/C/
[root@dds2 ~]# cd /opt/RTI/ndds.5.1.0/example/C/
[root@dds2 C]# ls
Distributed_Logger  Hello_dynamic  helloWorldPersistence  News
Hello_builtin      Hello_idl      helloWorldRequestReply
Hello_builtin_tcp  Hello_simple   helloWorldTCP
[root@dds2 C]#
```

Lista de ejemplos de RTI.

3. En el interior del ejemplo de aplicación Hello_simple se encuentran directorios siendo uno de los principales el llamado make, este tiene varios programas que hacen referencia a esta misma aplicación pero cada uno de estos funciona para cierto tipo de procesador.

Es necesario saber cuál es el procesador que tiene nuestra Pc antes de llevar a cabo una compilación de Hello_simple.

Para esto es necesario digitar el siguiente comando:

```
ls /$NDDSHOME/lib
```

El comando mostrara la versión instalada de RTI para el procesador de la Pc en cuestión.

La siguiente figura muestra los archivos de la carpeta make en la que debe estar el ejemplo Hello_simple que se compilara. Este debe coincidir con el formato del procesador de la Pc (i86Linux2.6gcc4.4.5).

```

root@dds2:/opt/RTI/ndds.5.1.0/example/C/Hello_simple/make
[root@dds2 lib]# cd ..
[root@dds2 ndds.5.1.0]# cd example/C/Hello_simple/
[root@dds2 Hello_simple]# cd make/
[root@dds2 make]# ls
Makefile.64p5AIX5.3xlc9.0
Makefile.64p7AIX7.1xlc12.1
Makefile.armv6vfpLinux3.xgcc4.7.2
Makefile.armv7aQNX6.5.OSP1gcc_cpp4.4.2
Makefile.armv7leLinux2.6gcc4.4.1
Makefile.cell64Linux2.6gcc4.5.1
Makefile.cell64Linux2.6ppu4.1.1
Makefile.common
Makefile.i86Linux2.4gcc3.2
Makefile.i86Linux2.4gcc3.2.2
Makefile.i86Linux2.6gcc3.4.3
Makefile.i86Linux2.6gcc3.4.5
Makefile.i86Linux2.6gcc3.4.6
Makefile.i86Linux2.6gcc4.1.1
Makefile.i86Linux2.6gcc4.1.2
Makefile.i86Linux2.6gcc4.2.1
Makefile.i86Linux2.6gcc4.3.4
Makefile.i86Linux2.6gcc4.4.3
Makefile.i86Linux2.6gcc4.4.5
Makefile.i86Linux3gcc4.3.4
Makefile.i86Linux3.xgcc4.6.3
Makefile.i86QNX6.4.0gcc_acpp4.2.4
Makefile.i86QNX6.4.1gcc_gpp
Makefile.i86QNX6.5gcc_gpp4.4.2
Makefile.i86RedHawk5.1gcc4.1.2
Makefile.i86RedHawk5.4gcc4.2.1
Makefile.i86Sol2.10gcc3.4.4
Makefile.i86Sol2.9gcc3.3.2
Makefile.i86Suse10.1gcc4.1.0
Makefile.i86Suse9.5gcc3.3.3
Makefile.p5AIX5.3xlc9.0
Makefile.power7Linux2.6gcc4.4.4
Makefile.ppc4xxFPLinux2.6gcc4.3.3
Makefile.ppc4xxFPLinux2.6gcc4.5.1
Makefile.ppc7400Linux2.6gcc3.3.3
Makefile.ppc85xxLinux2.6gcc4.3.2
Makefile.ppcbeQNX6.4.0gcc_acpp4.2.4
Makefile.ppcLinux2.6gcc4.1.2
Makefile.sparc64Sol2.10cc5.8
Makefile.sparc64Sol2.10gcc3.4.2
Makefile.sparc64Sol2.9gcc3.2
Makefile.sparcSol2.10gcc3.4.2
Makefile.sparcSol2.8cc5.2
Makefile.sparcSol2.8gcc3.2
Makefile.sparcSol2.9cc5.3
Makefile.sparcSol2.9cc5.4
Makefile.sparcSol2.9gcc3.2
Makefile.sparcSol2.9gcc3.3
Makefile.x64Darwin10gcc4.2.1
Makefile.x64Darwin12clang4.1
Makefile.x64Linux2.4gcc3.2.3
Makefile.x64Linux2.6gcc3.4.5
Makefile.x64Linux2.6gcc4.1.1
Makefile.x64Linux2.6gcc4.1.2
Makefile.x64Linux2.6gcc4.3.2
Makefile.x64Linux2.6gcc4.3.4
Makefile.x64Linux2.6gcc4.3.4
Makefile.x64Linux2.6gcc4.4.3
Makefile.x64Linux2.6gcc4.4.4
Makefile.x64Linux2.6gcc4.4.5
Makefile.x64Linux2.6gcc4.5.1
Makefile.x64Linux3.xgcc4.6.3
Makefile.x64Sol2.10gcc3.4.3
Makefile.x64Suse10.1gcc4.1.0
Makefile.x64Suse9.5gcc3.3.3
[root@dds2 make]#

```

Archivos de compilación del programa DDS para diferentes versiones

4. Luego, se procederá a realizar la compilación del programa Hello_simple, para esto debemos estar ubicados en la ruta:

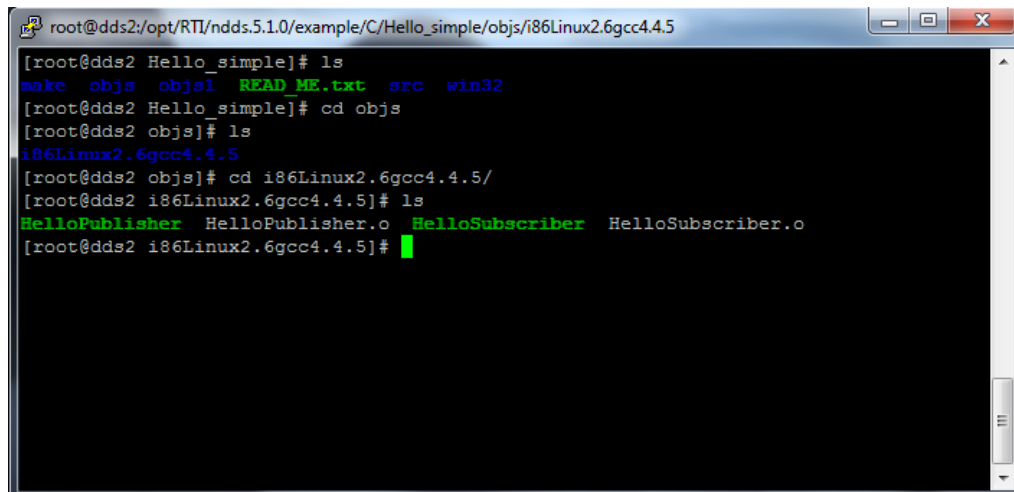
```
cd /$NDDSHOME/example/C/Hello_simple/
```

El comando para compilar Hello_simple es el siguiente:

```
make -f make/Makefile.i86Linux2.6gcc4.4.5 DEBBUG 1
```

5. Este comando crea la carpeta objs dentro del mismo directorio Hello_simple la cual contiene las carpetas de HelloPublisher y HelloSubscriber para las funciones de publicador y suscriptor respectivamente.

Para comprobar esto se puede escribir ls como muestra la siguiente imagen.

A terminal window with a blue title bar containing the path 'root@dds2:/opt/RTI/ndds.5.1.0/example/C/Hello_simple/objs/i86Linux2.6gcc4.4.5'. The terminal text shows the following sequence of commands and outputs:

```
[root@dds2 Hello_simple]# ls
make objs objsi READ ME.txt src win32
[root@dds2 Hello_simple]# cd objs
[root@dds2 objs]# ls
i86Linux2.6gcc4.4.5
[root@dds2 objs]# cd i86Linux2.6gcc4.4.5/
[root@dds2 i86Linux2.6gcc4.4.5]# ls
HelloPublisher HelloPublisher.o HelloSubscriber HelloSubscriber.o
[root@dds2 i86Linux2.6gcc4.4.5]#
```

Archivos compilados para ejecución de DDS.

6. Como siguiente paso debemos ejecutar HelloPublicador de la siguiente manera:

```
[root@dds2 i86Linux2.6gcc4.4.5]# ./HelloPublisher
```

Luego de esto es posible que la ejecución del ejemplo genere un error por la licencia, en este caso para solucionar esto se requiere copiar el archivo de la licencia license.dat que se menciono anteriormente al directorio /RTI/ndds.5.1.0.

Se puede realizar este paso de la siguiente manera:

```
cp /home/opt/license.dat /opt/RTI/ndds.5.1.0
```

Este comando copia licencia.dat de /opt a /ndds.5.1.0

Realizado esto es posible completar de manera correcta la ejecución de HelloPublisher .

7. A continuación abrir otro terminal para comprobar de esta manera el funcionamiento tanto del publicador como del suscriptor.

Es el nuevo terminal digitamos:

```
cd /opt/RTI/ndds.5.1.0/example/C/Hello_simple/objs/i86Linux2.6gcc 4.4.5/
```

```
./HelloSuscriber
```

En la siguiente figura se observa la ejecución de HelloPublisher y HelloSuscriber en dos terminales.

The image shows two terminal windows side-by-side. The top window is titled 'root@dds2:/opt/RTI/ndds.5.1.0/example/C/Hello_simple/objs/i86Linux2.6gcc4.4.5'. It shows the following commands and output:
[root@dds2 ~]# cd /opt/RTI/ndds.5.1.0/example/C/Hello_simple/objs/i86Linux2.6gcc4.4.5/
[root@dds2 i86Linux2.6gcc4.4.5]# ls
HelloPublisher HelloSubscriber HelloSubscriber.o
[root@dds2 i86Linux2.6gcc4.4.5]# ./HelloPublisher
RTI Data Distribution Service University License issued to Universidad de Chile - CHILE01-UP mmatus@centroenergias.cl For non-production use only.
Expires on 03-jun-2015 See www.rti.com for more information.
Ready to write data.
When the subscriber is ready, you can start writing.
Press CTRL+C to terminate or enter an empty line to do a clean shutdown.
Please type a message> []

The bottom window is titled 'root@dds2:/opt/RTI/ndds.5.1.0/example/C/Hello_simple/objs/i86Linux2.6gcc4.4.5'. It shows the following commands and output:
[root@dds2 ~]# cd \$NDDSHOME
[root@dds2 ndds.5.1.0]# cd example/C/Hello_simple/objs/i86Linux2.6gcc4.4.5/
[root@dds2 i86Linux2.6gcc4.4.5]# ls
HelloPublisher HelloSubscriber HelloSubscriber.o
[root@dds2 i86Linux2.6gcc4.4.5]# ./HelloSubscriber
RTI Data Distribution Service University License issued to Universidad de Chile - CHILE01-UP mmatus@centroenergias.cl For non-production use only.
Expires on 03-jun-2015 See www.rti.com for more information.
Ready to read data.
Press CTRL+C to terminate.
[]

Publicador y suscriptor para pruebas funcionales

8. Con estos dos terminales se podrá observar el funcionamiento de DDS escribiendo datos para transmitir en el publicador y recibiendo estos por el suscriptor. Estos ejemplos se pueden realizar entre distintas computadoras y entre distintas Single boards.

ANEXO B

TUTORIAL PARA LA INSTALACIÓN DE DDS DE RTI EN RASPBERRY PI.

Contenido

TUTORIAL PARA LA INSTALACIÓN DE DDS DE RTI EN RASPBERRY PI.....	81
1. Instalación de Raspbian.....	81
2. Descarga de DDS de RTI.....	83
3. Instalación de DDS de RTI en Raspberry Pi.....	83
4. Configuración de DDS de RTI.....	85
5. Pruebas de DDS de RTI en la Raspberry.....	86

Para la instalación de DDS en el single-board computer Raspberry Pi se recomienda en primer lugar tener instalado el sistema operativo Raspbian que es el controlador básico para el manejo del single board en el que se podrá instalar todos los componentes para la correcta función de DDS.

La instalación de DDS es un tanto diferente a la instalación típica en una Pc ya que el single board tiene un sistema operativo denominado Raspbian.

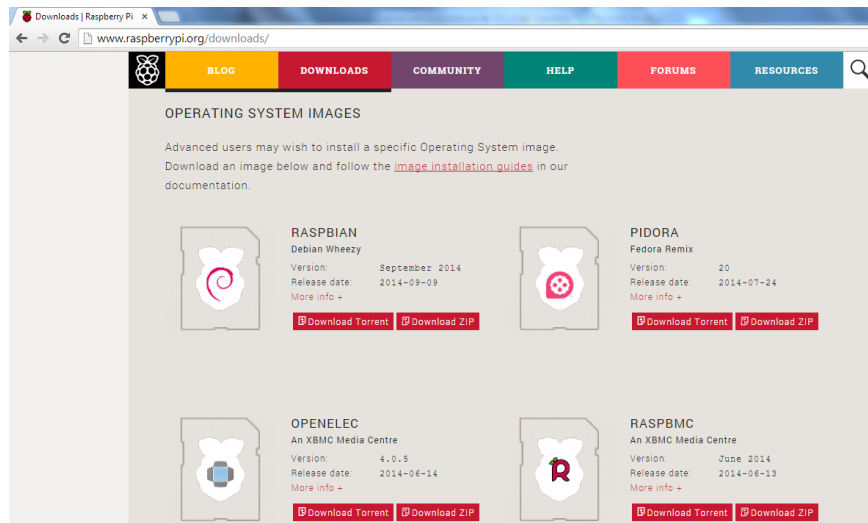
1. Instalación de Raspbian.

Los pasos para la instalación de Raspbian en el disco duro de la Rasp (Memoria SD, MicroSD) son sencillos, ya que se necesita tan solo el programa Raspbian y un programa que escriba el sistema operativo en la Memoria. Es necesario disponer de una memoria de al menos 8Gb de almacenamiento.

Raspbian se puede descargar desde el siguiente sitio:

<http://www.raspberrypi.org/downloads/>

La siguiente imagen muestra el sitio web con algunos de los sistemas operativos disponibles en la pagina web.



Sitio web para la descarga del sistema operativo Raspbian

Actualización y configuración de Raspbian.

Antes de la descarga de DDS de RTI es conveniente actualizar la Raspberry Pi. Para la actualización completa se debe digitar los tres siguientes comandos:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo rpi-update
```

Adicionalmente se debe expandir la memoria para que se pueda utilizar todo el espacio de la memoria SD como disco duro para raspbian, para esto es necesario ingresar en las configuraciones del sistema con el siguiente comando:

```
sudo raspi-config
```

Luego ingresar en expandir memoria y seleccionar la opción correcta.

Una de las configuraciones importantes que se debe realizar a la Raspberry es aumentar la memoria RAM y SWAP para el buen funcionamiento de las aplicaciones que se ejecuten.

Se debe modificar el siguiente archivo para aumentar la RAM:

```
sudo nano /etc/sysctl.conf
```

buscar la línea `vm.min_free_kbytes` que tiene un valor 8192 por defecto y aumentarla a 16384 como mínimo

```
vm.min_free_kbytes = 16384
```

También se debe modificar el siguiente archivo para aumentar la memoria SWAP:

```
sudo nano /etc/dphys-swapfile
```

Quedando de la siguiente manera

```
CONF_SWAPSIZE=1024
```

Es necesario reiniciar para aplicar los cambios con el comando:

```
sudo init 6
```

Tras el reinicio podemos comprobar el tamaño con el siguiente comando.

```
free -h
```

2. Descarga de DDS de RTI

Antes de la descarga de DDS de RTI es conveniente actualizar la Raspberry Pi. Para la actualización completa se debe digitar los tres siguientes comandos:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo rpi-update
```

Luego de la actualización del sistema operativo Raspbian se procederá a descargar el programa DDS de RTI del siguiente sitio web:

```
http://www.rti.com/downloads/connext-files.html#DOWNLOAD
```

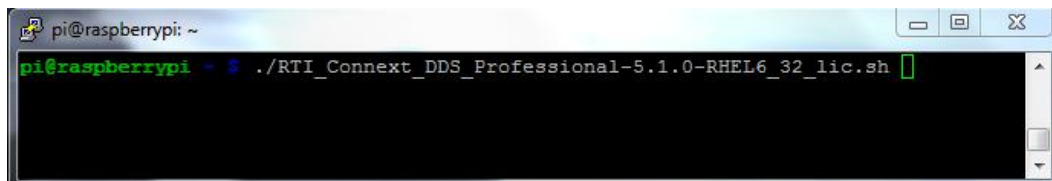
Es necesario descargar el software DDS de RTI de 32bits directamente desde la Rasp con el siguiente comando:

```
wget http://s3.amazonaws.com/RTI/eval/rtidds510/RTI_Connext_DDS_Professional-5.1.0-RHEL6_32_lic.sh
```

3. Instalación de DDS de RTI en Raspberry Pi.

Para instalar DDS de RTI en la Rasp digitamos el comando:

```
./RTI_Connext_DDS_Professional-5.1.0-RHEL6_32_lic.sh
```



Comando para la instalación de DDS.

RTI Connex DDS Profesional originalmente está dispuesto para un procesador i86Linux2.6gcc4.4.5 y es compatible con una Pc de escritorio o portátil. La Rasp tiene un tipo de procesador armv6vfpLinux3.xgcc4.7.2 por lo que es necesario descargar las librerías correspondientes para el buen funcionamiento de DDS en la Raspberry Pi.

Single Board tiene un tipo de procesador (armv6) diferente al configurado por defecto en el DDS descargado de RTI que trabaja con un procesador diferente (i86).

Descarga de librerías para la Raspberry Pi

Como siguiente paso tenemos que descargar las librerías para que DDS funcione correctamente en la Rasp.

El comando para descargar las librerías es el siguiente:

```
mkdir /home/pi/lib_DDS
```

```
wget http://s3.amazonaws.com/RTI/Community/ports/RTI_Connext_DDS_Core_Target-5.1.0-armv6vfpLinux3.xgcc4.7.2.tar.gz
```

Es necesario descomprimir el archivo descargado con el siguiente comando:

```
tar -xvzf RTI_Connext_DDS_Core_Target-5.1.0-armv6vfpLinux3.xgcc4.7.2.tar.gz
```

El comando anterior genera una carpeta con el nombre ndds.5.1.0. que contiene las siguientes carpetas: class, lib y el archivo rev_target_rtidds.5.1.0. La carpeta lib esta debe ser reemplazada por la carpeta lib del mismo nombre que se encuentra en la dirección: /home/pi/RTI/ndds5.1.0

Para realizar esto se pueden digital los comandos:

```
pi@raspberrypi ~ $ mkdir lib_old
```

```
pi@raspberrypi ~ $ cd /home/pi/RTI/ndds.5.1.0
```

```
pi@raspberrypi ~/RTI/ndds.5.1.0 $ mv lib /home/pi/lib_old
```

```
pi@raspberrypi ~/RTI/ndds.5.1.0 $ cd
```

```
pi@raspberrypi ~ $ cd /home/pi/lib_DDS/ndds.5.1.0
```

```
pi@raspberrypi ~/lib_DDS/ndds.5.1.0 $ mv lib /home/pi/RTI/ndds.5.1.0
```

También conviene copiar el archivo rev_target_rtidds.5.1.0. a la misma dirección menos el archivo cat por lo que ndds5.1.0 ya contiene esta carpeta con un contenido similar. Para realizar esto digitamos:

```
pi@raspberrypi ~/lib_DDS/ndds.5.1.0 $ mv rev_target_rtidds.5.1.0 /home/pi/RTI/ndds.5.1.0
```

Se recomienda también cambiar los permisos de este archivo escribiendo:

```
chmod 777 rev_target_rtidds.5.1.0
```

Descarga de JDK 8

A continuación se procederá también a descargar la versión 8 de JDK para actualizar la versión de java instalada en la Rasp.

Para actualizar JDK de java es necesario primero descargarlo, para esto se debe digitar el comando:

```
mkdir /home/pi/apps
```

Luego:

```
wget --no-check-certificate http://www.java.net/download/jdk8/archive/b98/binaries/jdk-8-ea-b98-linux-arm-vfp-hflt-10_jul_2013.tar.gz
```

Luego debemos descomprimirlo con el comando:

```
tar xvfz jdk-8-ea-b98-linux-arm-vfp-hflt-10_jul_2013.tar.gz
```

4. Configuración de DDS de RTI.

Como siguiente paso es necesario escribir las variables de estado dentro del archivo con el siguiente comando:

```
nano /home/pi/.bashrc
```

dentro de este archivo se deberá colocar la información siguiente:

```
export NDDSHOME=/home/pi/RTI/ndds.5.1.0
```

```
export PATH=$NDDSHOME/scripts:/home/pi/apps/jdk1.8.0/lib:$PATH
```

```
export
```

```
LD_LIBRARY_PATH=$NDDSHOME/lib/armv6vfpLinux3.xgcc4.7.2jdk:$LD_LIBRARY_PATH
```

La siguiente grafica muestra cómo debe quedar escrito el archivo .bashrc

```

pi@Rasp2: ~
GNU nano 2.2.6                                Fichero: .bashrc
# sources /etc/bash.bashrc).
if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
  . /etc/bash_completion
fi

export NDDSHOME=/home/pi/RTI/ndds.5.1.0
export PATH=$NDDSHOME/scripts:/home/pi/apps/jdk1.8.0/bin:$PATH
export LD_LIBRARY_PATH=$NDDSHOME/lib/armv6vfpLinux3.xgcc4.7.2jdk:$LD_LIBRARY_PATH

^G Ver ayuda      ^C Guardar      ^R Leer Fich    ^Y Pág Ant
^X Salir         ^J Justificar   ^W Buscar      ^V Pág Sig

```

Modelo de Archivo .bashrc luego de las configuraciones.

Como siguiente paso para la configuración de DDS se deberá ejecutar el programa rtiddsspy.

Al correr el programa rtiddsspy se activa el DDS de RTI para poder correr los ejemplos que trae incluido el software y debe mostrar la información como muestra la siguiente figura.

Nota: antes de ejecutar el programa rtiddsspy es necesario que la Pc este conectada a una red para en esta realizar luego las pruebas y que conecte a internet.

```

pi@raspberrypi: ~/RTI/ndds.5.1.0/scripts
pi@raspberrypi ~/$ cd RTI/ndds.5.1.0/scripts/
pi@raspberrypi ~/RTI/ndds.5.1.0/scripts $ ls
jms2dds      nddsspy      rtiddsping   rtipersistenceservice
ndds2gen     rtiddsgen    rtiddspython rtiddsprototyper
ndds2gen     rtiddsgen2   rtiddsspy    rtiddsprototyper
nddsprototyper rtiddsgen2_server rtiddsversion
pi@raspberrypi ~/RTI/ndds.5.1.0/scripts $ ./rtiddsspy

RTI Data Distribution Service Spy built with DDS version: 5.1.0 (Core: 1.7a.0
0, C: 1.7a.00, C++: 1.7a.00)
Copyright 2012 Real-Time Innovations, Inc.
~~~~~
NddsSpy is listening for data, press CTRL+C to stop it.

source_timestamp  Info  Src HostId  topic  type
-----

```

Ejecución del programa rtiddsspy

5. Pruebas de DDS de RTI en la Raspberry.

Una vez terminado de descargar, instalar y configurar DDS se puede proceder a realizar las primeras pruebas para comprobar el correcto funcionamiento de envío y recepción de datos.

Hay que recordar que el funcionamiento de DDS se basa en publicadores que envían la información hacia la red y suscriptores que recogen o toman esa información para procesarla. Un mismo terminal puede funcionar como publicador y suscriptor si se desea.

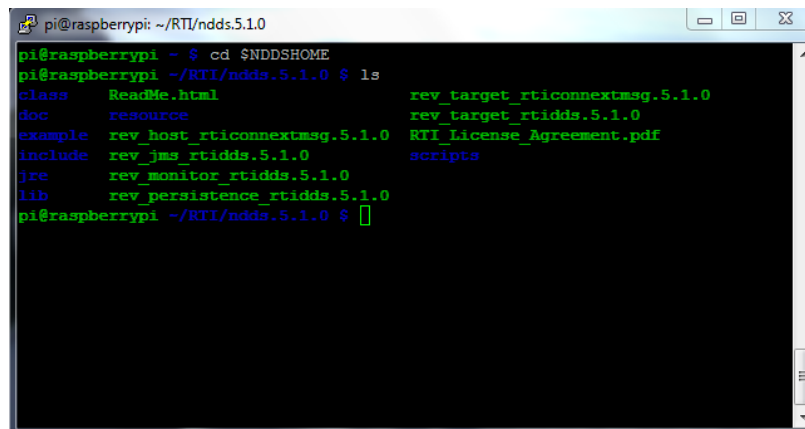
Uno de los ejemplos de prueba que trae DDS de RTI es Hello_simple el cual se podrá ejecutar un suscriptor y un publicador con lo que se dará la posibilidad de un intercambio de datos en la red.

El primer paso para ejecutar el programa Hello es la compilación del mismo, para ello es necesario haber configurado correctamente las variables de entorno y el DDS.

La compilación del ejemplo se realiza de la siguiente manera:

Nos ubicamos bajo el directorio /home/pi/ndds.5.1.0 para ello digitamos:

```
cd $NDDSHOME
```



```
pi@raspberrypi: ~/RTI/ndds.5.1.0
pi@raspberrypi ~ $ cd $NDDSHOME
pi@raspberrypi ~/RTI/ndds.5.1.0 $ ls
class  README.html          rev_target_rticonnextmsg.5.1.0
doc    resource             rev_target_rtids.5.1.0
example rev_host_rticonnextmsg.5.1.0  RTI_License_Agreement.pdf
include rev_jms_rtids.5.1.0      scripts
jre    rev_monitor_rtids.5.1.0
lib    rev_persistence_rtids.5.1.0
pi@raspberrypi ~/RTI/ndds.5.1.0 $
```

Archivos que debe contener la carpeta ndds.5.1.0

Como se observa en la figura el comando direcciona hacia /home/pi/RTI/ndds.5.1.0, esto quiere decir que la variable de entorno NDDSHOME está bien definida.

Luego es necesario digitar el siguiente comando para ir a la ubicación de Hello_simple:

```
pi@raspberrypi ~/RTI/ndds.5.1.0 $ cd example/C/Hello_simple/
```

En esta carpeta existe una carpeta make la cual tiene las versiones del programa Hello_simple para cada tipo de procesador, en este caso el archivo que se compilará será el Makefile.armv6vfpLinux3.xgcc4.7.2 que es compatible con la Raspberry.

Para la compilación escribimos

```
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple $ make -f
make/Makefile.armv6vfpLinux3.xgcc4.7.2
```

La compilación por si misma genera una carpeta llamada objs que se encuentra en la dirección:

```
/home/pi/RTI/ndds.5.1.0/example/C/Hello_simple
```

La siguiente figura muestra la carpeta objs y su contenido.

```
pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2
pi@raspberrypi ~ $ cd $NDDSHOME
pi@raspberrypi ~/RTI/ndds.5.1.0 $ cd example/C/Hello_simple/
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple $ ls
make  objs  READ_ME.txt  src  win32
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple $ cd objs/
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs $ ls
armv6vfpLinux3.xgcc4.7.2
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs $ cd armv6vfpLinux3.xgcc4.7.2/
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2 $ ls
HelloPublisher  HelloPublisher.o  HelloSubscriber  HelloSubscriber.o
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2 $
```

Contenido de la carpeta obj de Hellosimple

Como podemos ver en la imagen dentro de la carpeta obj se encuentra otra carpeta llamada armv6vfpLinux3.xgcc4.7.2 que en su interior contiene el programa publicador (HelloPublisher) y suscriptor (HelloSubscriber) respectivamente.

Para probar este ejemplo es necesario abrir otra terminal para que en el primer terminal ejecutar HelloPublisher y en el segundo terminal ejecutar HelloSubscriber. La siguiente figura muestra como se debe ejecutar los dos ejemplos.

```
pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2
pi@raspberrypi ~ $ cd $NDDSHOME
pi@raspberrypi ~/RTI/ndds.5.1.0 $ cd example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2/
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2 $ ls
HelloPublisher  HelloPublisher.o  HelloSubscriber  HelloSubscriber.o
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2 $ ./HelloPublisher
Ready to write data.
When the subscriber is ready, you can start writing.
Press CTRL+C to terminate or enter an empty line to do a clean shutdown.
Please type a message>

pi@raspberrypi: ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2
pi@raspberrypi ~ $ cd $NDDSHOME
pi@raspberrypi ~/RTI/ndds.5.1.0 $ cd example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2/
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2 $ ls
HelloPublisher  HelloPublisher.o  HelloSubscriber  HelloSubscriber.o
pi@raspberrypi ~/RTI/ndds.5.1.0/example/C/Hello_simple/objs/armv6vfpLinux3.xgcc4.7.2 $ ./HelloSubscriber
Ready to read data.
Press CTRL+C to terminate.
```

Publicador y suscriptor listos para pruebas funcionales.

Realizado esto está todo listo para probar la transmisión de datos desde HelloPublisher a HelloSubscriber y se puede digitar cualquier dato en el terminal superior que el respectivo suscriptor tomara los datos y los imprimirá en la pantalla del suscriptor.

ANEXO C

CÓDIGO C/C++ DEL CÁLCULO Y TRANSMISIÓN DE LA FRECUENCIA EN RASPBERRY PI

```
//Include arduPi library
#include "arduPi.h"           //libreria descargable propia de la tarjeta del conversor A/D.
#include "time.h"

unsigned char val_0 = 0;
unsigned char val_1 = 0;
byte address = 0x08;

int channel_0 = 0;
double analog_0 = 0.0;
struct timespec last;
struct timespec now;
struct timespec mid;
int ctr;
double last_analog_0 = 0.0;
double last_time = 0.0;
long t_now = 1.0;
long t_nnow = 1.0;
double t_last = 0.0;
double diff = 1.0;
double delta = 1.0;
double thresh = 2.5;
int on = 1;
double last_freq = 0.0;
double avg_freq = 0.0;
double freq = 0.0;
double freq1 = 50.0;
char output[10];
double m = 0.0;
double b1 = 0.0;
double b2 = 0.0;
double t_now_sec = 0.0;
double delta_sec = 0.0;
double t_last_sec = 0.0;
double t_inter= 0.0;
long t_mid =0.0;
long t_nmid =0.0;
```

```

double delta_mid =0.0;
double t_mid_sec = 0.0;

void setup()
{
  Wire.begin();                //habilita el conversor A/D
  ctr = 1;
  clock_gettime(CLOCK_REALTIME, &now); //habilita el reloj interno de tiempo real
  clock_gettime(CLOCK_REALTIME, &mid); //habilita el reloj interno de tiempo real
}

void loop()                    //bucle principal del programa
{
  // channel 0
  Wire.beginTransaction(8);     //habilita canal 0 del convesor A/D
  Wire.write(byte(0xDC));      //Envía la dirección del canal 0

  char val[2];
  val_0 = Wire.read();         //recibe byte alto (sobrescribe lectura previa)
  val_1 = Wire.read();
  channel_0 = int(val_0)*16 + int(val_1>>4);
  analog_0 = channel_0 * 5.0 / 4095.0;

  if (analog_0 >= thresh){     //primera condición - si lectura es mayor a umbral
    if (last_analog_0 < thresh){ //segunda condición - si última lectura es menor a umbral
      clock_gettime(CLOCK_REALTIME, &now);
      t_now=now.tv_sec;        //captura de tiempos en variables
      t_nnow= now.tv_nsec;     //captura de tiempos en variables
      t_mid =mid.tv_sec;       //captura de tiempos en variables
      t_nmid =mid.tv_nsec;     //captura de tiempos en variables
      t_now_sec=(double)t_now+(double)((double)t_nnow)/1000000000;
      t_mid_sec=(double)t_mid+(double)((double)t_nmid)/1000000000;

      delta_mid=t_now_sec - t_mid_sec; //calculo de la delta de tiempo

      m = (analog_0-last_analog_0)/delta_mid;
      b1 = analog_0-m*t_now_sec;
      b2 = last_analog_0 - m*t_last_sec;
      t_inter = (thresh - b1)/m;      //interpolación de datos
      delta_sec=t_inter - t_last_sec;
      freq = 1/delta_sec;             //cálculo de la frecuencia
      t_last_sec = t_inter;
    }
  }
}

```

```

        if (freq >= 40) //límite inferior de frecuencia
            if (freq <= 70) //límite superior de frecuencia
                {
                    snprintf(output,10,"%f",freq);
                    printf("%s\n",output); //imprime en pantalla la frecuencia
                    fflush(stdout);
                }
            }
    freq, analog_0, diff, t_now, delta, t_last, t_now);
    }
}
clock_gettime(CLOCK_REALTIME, &mid);
last_analog_0 = analog_0;
}

int main (){
    setup(); //llama a void setup
    printf("Initiating loop sequence.\n");
    while(on){
        loop(); //llama a bucle principal del programa
    }
    return (0);
}

```