

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y Electrónica



TRABAJO DE DIPLOMA

**Diseño de una unidad de adquisición de datos
para la medición de la frecuencia en una red de
cobre presurizada**

Autor: Mercedes Fernández Díaz.

Tutor: Ing. Samy Brito Barroso.

Consultante: Ing. Alexeis Montero Fonstecilla.

Santa Clara

2015

"Año 57 de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y Electrónica



TRABAJO DE DIPLOMA

**Diseño de una unidad de adquisición de datos
para la medición de la frecuencia en una red de
cobre presurizada**

Autor: Mercedes Fernández Díaz.

E-mail: mercy.fdez @nauta.cu

Tutor: Ing. Samy Brito Barroso.

Prof. Asistente, Departamento de Automática y Sistemas Computacionales Facultad de Ingeniería Eléctrica UCLV.

E-mail: Samyb @uclv.edu.cu

Consultante: Ing. Alexeis Montero Fonstecilla.

Santa Clara

2015

"Año 57 de la Revolución"



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Telecomunicaciones, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma del Jefe de Departamento
donde se defiende el trabajo

Firma del Responsable de
Información Científico-Técnica

PENSAMIENTO

“Lo más auténtico de nosotros es nuestra capacidad de crear, de sobreponernos, de soportar, de transformar, de amar, de ser más grandes que nuestro sufrimiento....”

DEDICATORIA

A mi hermana por confiar siempre en mí, por ser mi cómplice de tantas alegrías.....

AGRADECIMIENTOS

Especialmente a mi hermana para, que veas chiquita que cuando la vida nos golpea debemos tener corazón para devolverle el golpe.

A mi mamá por su comprensión, desvelos y sacrificios para hacer realidad este sueño compartido.

A mi papa por impulsarme a seguir luchando para llegar sonriente a la meta, por convocarme a sus travesuras que me hacen tan feliz.

Colocar en la cúspide de esta gran victoria a toda mi familia sin ellos a mi lado no hubiese sido posible.

A Ale que más que mi tutor ha sido mi amigo, mi guía, gracias por acoger la realización de este trabajo como tuyo propio, muchas gracias de todo corazón por tanta paciencia y dedicación, por haber soportado mis desesperos estos años.

Al profesor Samy por su tutoría admirable, por su voluntad de ayudarme y por creer en mí.

A mis amigos de toda la vida y a tantas personas que a lo largo de estos años me han tendido la mano de una u otra manera, a todo los que sin importar me han apoyado, muchas gracias.

TAREA TÉCNICA

Diseñar una unidad de adquisición de datos empleando la plataforma Arduino, basada en microcontroladores, que permita captar los valores de los transductores de presión de los cables de cobre presurizados de ETECSA.

Firma del Autor

Firma del Tutor

RESUMEN

La presurización de cables telefónicos constituye hoy el método más utilizado para preservar los parámetros eléctricos de los pares conductores, su implementación debe ser complementada con métodos apropiados de monitoreo y gestión. La adquisición de los valores de presión en el interior de las corazas se hace a través de transductores de presión que se conectan eléctricamente utilizando cables telefónicos o por medio del Sistema Remoto de Adquisición de Datos de la Presurización a una Unidad de Adquisición. El alto costo de este equipamiento dificulta su presencia en todos los sitios por lo que en este trabajo se propone una alternativa que posibilita resolver este problema con un costo inferior, para así formar una unidad de adquisición alternativa que permita la obtención de datos asociados a la presurización.

TABLA DE CONTENIDOS

PENSAMIENTO	I
DEDICATORIA	II
AGRADECIMIENTOS	III
TAREA TÉCNICA.....	IV
RESUMEN.....	V
TABLA DE CONTENIDOS	VI
INTRODUCCIÓN	1
CAPÍTULO 1. Aspectos generales de presurización.....	4
1.1. Historia de los sistemas de presurización.....	4
1.2. Objetivos y ventajas de la presurización.....	6
1.2.1. Limitaciones.....	8
1.3. Componentes de un sistema de presurización.....	8
1.3.1. Fuente de alimentación.....	9
1.3.2. Panel de distribución.....	10
1.3.3. Otros componentes del sistema de presurización.....	10
1.4. Sistemas de monitoreo y gestión de la presurización.....	11
1.4.1. Transductores	12
1.4.2. Unidades de adquisición de datos.....	12
1.4.3. Unidad Central de Procesamiento.....	12
1.5. Sistemas de monitoreo y gestión de la presurización en el mundo.....	13
1.6. Estado de la presurización en ETECSA.....	14
1.6.1. Estado de los sistemas de monitorización y gestión de la presurización en ETECSA.....	14

1.7. Consideraciones finales del capítulo.....	17
CAPÍTULO 2. Implementación de la unidad de adquisición de datos.....	18
2.1. Características técnicas y funcionamiento de los transductores de presión TP204M std.	18
2.2. Métodos de medición de frecuencia.....	21
2.2.1. Contadores recíprocos.....	22
2.3. Plataforma Arduino.....	23
2.3.1. Origen del proyecto Arduino.....	24
2.3.2. Fundamentos del proyecto Arduino.....	24
2.3.3. El microcontrolador y la placa.....	26
2.3.4. Expansiones o shields.....	27
2.4. Arduino Ethernet R3.....	27
2.4.1. Características técnicas generales:.....	28
2.4.2. Entradas y salidas.....	28
2.4.3. Comunicaciones.....	29
2.5. Inserción de la unidad de adquisición en el Sistema de gestión y monitoreo de ETECSA.....	30
2.6. Algoritmo para medir frecuencia.....	31
2.7. Algoritmo para la medición de las señales emitidas por los transductores.....	31
2.8. Conclusiones parciales.....	32
CAPÍTULO 3. Resultados.....	33
3.1. Medición de frecuencia con Arduino.....	33
3.1.1. Resultados obtenidos en la medición de frecuencia.....	34
3.2. Implementación del software para la unidad de adquisición de datos.....	36

3.3. Validación del algoritmo de medición para una línea de transductores de presión TP204M (std) mediante simulación por computadora.	36
3.3.1. Simulación de la línea de transductores.	37
3.3.2. Simulación de la unidad de adquisición de datos.	37
3.4. Valoración económica y medioambiental.	38
3.5. Conclusiones parciales del capítulo	38
Conclusiones.	39
Referencias bibliográficas.	41
Anexo 1 Datos técnicos del transductor de presión TP204M (std).	44
Anexo 2 Algoritmo para la medición de una línea de transductores TP204M (std) con la unidad de adquisición de datos.	46
Anexo 3 Software para medir frecuencia y registrar mediciones con Arduino Ethernet R3.	48
Anexo 4 Generador de funciones Leader LFG-1300S (extracto).	51
Anexo 5 Especificaciones de los contadores de la serie Nova.	53
Anexo 6 Resultados de las mediciones de frecuencia.	54
Anexo 7 Fuentes de incertidumbre total asociada al patrón de referencia.	61
Anexo 8 Programa para la unidad de adquisición de datos basado en Arduino Ethernet R3.	62
Anexo 9 Comprobación de los algoritmos de medición de la unidad de adquisición de datos alternativa.	67

INTRODUCCIÓN

Los servicios de telecomunicaciones se encuentran en un proceso de desarrollo constante, el uso de redes de fibra óptica logra una eficiencia superior a las de cobre, sin embargo su costo es mucho más elevado. Su utilización en Cuba está limitada a la red de transporte. El desarrollo de tecnologías RDSI (Red digital de servicios integrados) y xDSL (*x Digital Subscriber Line*) en sus múltiples variantes, ha permitido que el par de cobre no sólo sea utilizado para telefonía básica, hoy este servicio constituye uno más dentro de la gama de servicios que se ofrecen utilizando este medio, aprovechándose así la infraestructura instalada. Modernas tecnologías permiten un aumento del ancho de banda transmitido por los pares de cobre.

Para garantizar la calidad de los servicios es necesario conservar las características físicas y eléctricas de las líneas de transmisión. La degradación de estos parámetros ocasiona interrupciones que generalmente necesitan gran cantidad de tiempo y de recursos para su restablecimiento; afectando la cantidad y calidad de los servicios brindados.

La presurización de los cables telefónicos constituye el método más utilizado para preservar los parámetros eléctricos de los pares conductores, su implementación debe ser complementada con métodos apropiados de monitoreo y gestión.

La adquisición de los valores de presión en el interior de los cables se hace a través de transductores de presión que se conectan eléctricamente utilizando cables telefónicos a una unidad de adquisición de datos. En nuestro país, el alto costo de este equipamiento, en el mercado internacional, dificulta su utilización en todas las redes presurizadas (Benítez, 2012). El costo es una de las principales limitantes para la generalización del sistema de monitoreo, sobre todas en aquellas localidades donde las redes presurizadas son de pequeñas dimensiones.

La implementación de una unidad de adquisición de datos puede ser una alternativa que permita la obtención de los datos asociados a la presurización. Si se implementa una unidad de adquisición se puede extender el sistema de monitorio y gestión de la red de cables de cobre presurizados a todos los elementos de la red que actualmente no cuentan con el mismo.

El objetivo general de esta investigación es diseñar una unidad de adquisición de datos empleando la plataforma Arduino, que permita captar los valores de los transductores de presión de los cables de cobre presurizados de ETECSA. Para lograr este objetivo se establecen los siguientes objetivos específicos:

- Realizar un estudio del sistema de presurización y del sistema de monitoreo y gestión de la red de cables de cobre presurizados empleados en ETECSA.
- Investigar las características y aplicaciones de la plataforma Arduino.
- Elaborar el algoritmo que permita realizar un software sobre la plataforma Arduino para medir la frecuencia de los tonos de audio emitidos por los transductores de presión instalados en los cables presurizados.
- Medir los valores de presión captados por los transductores de presión.

Como resultado de esta investigación se espera obtener una unidad de adquisición de datos con un costo reducido, basada en tecnologías libres, flexible y adaptable. Este enfoque puede ser el inicio del desarrollo de unidades de adquisición más complejas y de un sistema de monitoreo y gestión propio que sustituya el que se utiliza actualmente en la Empresa de Telecomunicaciones de Cuba (ETECSA), disminuyendo los costos de instalación, mantenimiento y reparación de las actuales redes de cables de cobre existentes; que permitiría, en este campo, la independencia tecnológica tan necesaria en el contexto económico y social existente en el mundo actual.

El presente trabajo de diploma está estructurado en tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos. De manera resumida el contenido de los capítulos es el siguiente:

En el capítulo 1 se abordan aspectos generales de la presurización; los orígenes de los sistemas de presurización; los objetivos, ventajas y limitaciones en su implementación. Se describen los principales componentes de los sistemas presurizados, de los sistemas de monitoreo y gestión de la presurización y su estado en nuestro país.

En el capítulo 2 se hace referencia a los datos técnicos de los transductores de presión. Se abordan los métodos de medición de frecuencia, el método seleccionado para realizar este trabajo y sus principales ventajas. Las características técnicas, aplicaciones, herramientas y

ventajas del proyecto Arduino y del modelo Arduino Ethernet R3 seleccionado. Se muestra la inserción de la unidad de adquisición de datos implementada, en el sistema de gestión y monitoreo de la presurización en ETECSA, además las características generales de la interface necesaria para acondicionar la señal, recibida desde los transductores, de acuerdo a los requerimientos de entrada de la unidad de adquisición implementada. Se tratan los algoritmos para medir frecuencia y para medir las señales emitidas por los transductores de presión.

En el capítulo 3 se comenta sobre la medición de frecuencia con Arduino, y sobre los resultados obtenidos, sobre la validación del algoritmo de medición para una línea de transductores de presión TP204M mediante simulación por computadora incluyendo la simulación de la línea de transductores y de la unidad de adquisición de datos; además se realiza una valoración económica y medioambiental que ilustra el impacto de este trabajo.

Todos los capítulos terminan con un epígrafe de consideraciones finales en el que se resumen los aspectos más importantes que se trataron.

CAPÍTULO 1. Aspectos generales de presurización.

En el presente capítulo se realiza un análisis teórico de los sistemas de presurización, sus objetivos, ventajas y limitaciones. Además se hace un estudio de conceptos necesarios para una correcta comprensión del tema, se refieren sus principales componentes y los sistemas de monitoreo y gestión de estas redes utilizados en Cuba. Esta investigación permite conocer el desarrollo de tecnologías de avanzadas utilizadas a nivel mundial.

Este capítulo está estructurado en siete epígrafes. Se expone una breve reseña del desarrollo de los sistemas presurizados. Se tratan los objetivos, ventajas, limitaciones de la presurización y los componentes físicos de un sistema presurizado. Los sistemas de monitoreo y gestión de la presurización de manera general, los más utilizados en el mundo y el estado actual de la presurización en Cuba (ETECSA) son tratados en los epígrafes correspondientes a este capítulo, incluye además unas breves conclusiones.

1.1. Historia de los sistemas de presurización.

La presurización de cables telefónicos fue instituida por primera vez en la década del 40 del siglo XX, con el fin de proteger los cables telefónicos de la entrada de humedad (Plath, 2006).

Hasta finales de los años 70 la mayoría de los cables presurizados eran supervisados por medio de llaves de presión instaladas a lo largo del cable a intervalos prefijados. Cuando la presión descendía por debajo de un cierto umbral, se enviaba una alarma a la central telefónica.

Al detectarse una alarma el personal de mantenimiento debía enfrentarse a la siguiente situación (Puregas, 2002):

- Se suponía que la fuga debía encontrarse aproximadamente a la mitad de la distancia entre las llaves de presión anterior y posterior a la que resultase la alarma.
- Sin importar el momento el personal de mantenimiento debía partir a solucionar la falla.
- Se requerían de 3 a 4 días para reparar la avería.

Para lograr cambiar este modo de operación, comenzó la búsqueda de un sistema eficiente de monitoreo que detectara rápidamente cualquier anomalía e identificara precisamente y en tiempo la ubicación de la falla. Con el objetivo de proteger los cables telefónicos a partir de 1990 surgen los sistemas de monitoreo y gestión de la presurización, que basan su funcionamiento en la información ofrecida por transductores de presión instalados en los cables. El costo de una reparación cuando la falla en una coraza es localizada con precisión por un sistema de monitoreo y reparada antes de que los clientes sean afectados, comparado con el costo de reparación y el tiempo fuera de servicio si esta se humedece y se interrumpe, puede calcularse en cientos de miles de pesos como resultado directo de la eliminación de emergencias.

Aunque desde 1994 se extendió el uso de la presurización a todas las provincias del país, no fue hasta el año 2001 que se logró introducir el sistema de monitoreo y gestión de la presurización NiDA 1.15q y luego NiDA 2 desde el 2009, en las Direcciones Territoriales de La Habana, Matanzas, Villa Clara, Cienfuegos, Camagüey, Holguín y Santiago de Cuba, quedando aun sin ser monitoreados el 48.5 % de los cables presurizados (Yumar and Martínez, 2014).

El sistema actual de monitoreo y gestión exige la presencia de costosas unidades de adquisición para la obtención de los valores de presión en las corazas. En muchos casos esto constituye una traba importante sobre todo en aquellas regiones pequeñas donde no es factible económicamente la instalación de una Unidad de Adquisición.

En Cuba el monitoreo y gestión de la red de cables presurizados se ha abordado en diferentes trabajos de investigación realizados en universidades y en la Empresa de Telecomunicaciones de Cuba, siendo un tema no resuelto, la adquisición de los datos primarios (presiones) (Yumar and Martínez, 2014).

Para utilizar eficientemente el sistema instalado en ETECSA, el tema de la adquisición de datos primarios se aborda desde diversos enfoques:

Se diseña y construye por especialistas de ETECSA una tarjeta de control remoto cuya función es alimentar los transductores en el punto distante y transmitir la respuesta de los mismos hacia las Unidades de Adquisición (UA) instaladas en las Direcciones Territoriales

antes mencionadas. Con esta variante se garantiza poder instalar transductores en cualquier lugar y adquirir sus lecturas desde una UA distante independientemente del canal de transmisión utilizado (Benítez, 2012).

Otro acercamiento a la solución de esta problemática fue realizada en la Dirección Territorial de ETECSA de Camagüey, que consiste en utilizar una computadora para obtener las mediciones de presión de los transductores instalados. Esta solución se acerca más al funcionamiento de una unidad de adquisición. Se basa en una aplicación programada en MATLAB que a través de una interface con el par telefónico donde se encuentran instalados los transductores, obtiene las frecuencias emitidas por los mismos y mostrando los valores de presiones equivalentes (Yumar and Martínez, 2014).

A lo largo de los años se ha difundido el uso de la presurización para abarcar otros tipos de transportadores de comunicación tales como: cable Fibra Óptica, Cable coaxial y guías de ondas (Plath, 2006).

1.2. Objetivos y ventajas de la presurización.

Presurizar un cable telefónico consiste en inyectar un gas seco a presión en su interior (Figura 1.1), con el propósito de mantener dentro del mismo, una presión superior a la atmosférica y/o hidráulica a la que está sometido, evitando la penetración de humedad o agua en caso de presentar una falla en su cubierta protectora, o en algún cierre de empalme. Idealmente lo que se persigue es mantener a lo largo del cable una presión constante y que el consumo de gas sea cero (Pascual and Bocalandro, 2009).

La utilización de esta técnica en las empresas de telecomunicaciones ha demostrado que reduce la cantidad de averías en los circuitos, asegurando la continuidad del servicio y por consiguiente la disminución de las quejas de los abonados (Yumar and Martínez, 2014).

En muchas ocasiones el agua no logra introducirse por la cubierta de un cable. La humedad o vapor de agua contenido en el aire siempre penetra la cubierta al producirse succión capilar. La succión capilar es causada por los cambios de temperatura en el interior del cable (Belleza and Szymancyck, 2000).

El agua, con su alta constante dieléctrica altera la capacidad mutua entre pares y genera pérdidas dieléctricas que se incrementan conforme aumenta la frecuencia de la señal transmitida. El fenómeno de la electrólisis estará presente acentuando el deterioro (SEC-VAC, 2010). Mientras más humedad se absorbe más se degrada el aislamiento eléctrico. La circulación de gas seco, permite el secado permanente del interior del cable, conservando así intactas las características de aislamiento de estos y su vida útil. De esta forma se conservan los parámetros eléctricos de los cables. La presurización evita la sustitución de secciones enteras de cables y tener que abrir sus cubiertas, posibilitando que la localización y reparación de fallas puedan desarrollarse en el menor tiempo posible, mediante procedimientos sencillos, antes que los circuitos sufran interrupciones o se vea afectado el servicio (Jorge, 2011).

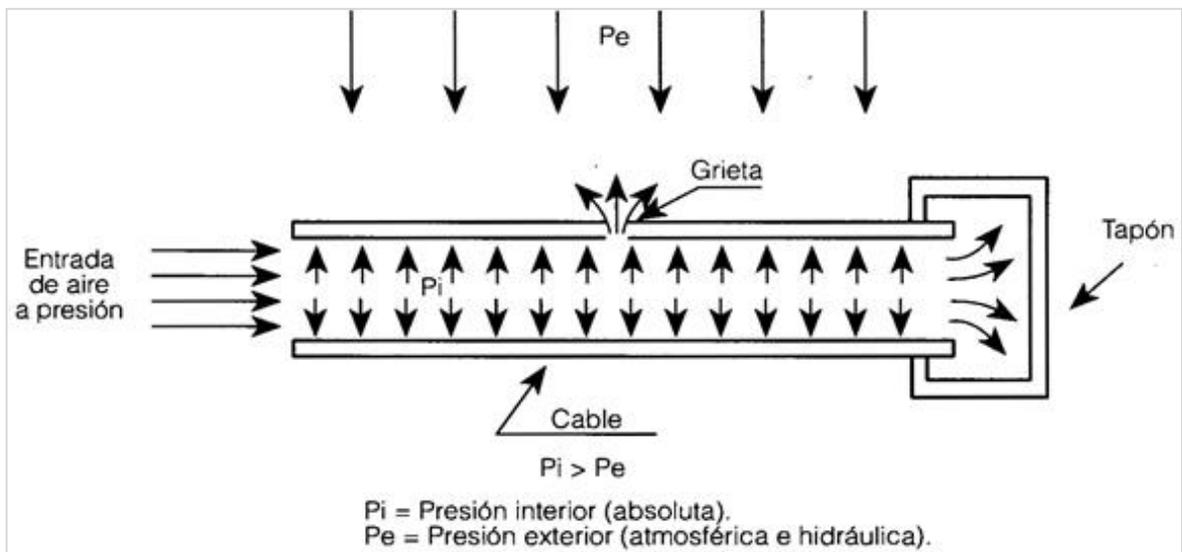


Figura 1.1. Esquema de un cable presurizado.

De esta forma aumenta la disponibilidad del sistema telefónico, con una sensible disminución en la interrupción del servicio y por consiguiente de la pérdida de ingresos por disminución del tráfico (Puregas, 2012).

Esta técnica logra una mejor organización de las labores de mantenimiento, enfocando la atención hacia las averías más urgentes. Las intervenciones correctivas relativas a fallas pequeñas, cuya dinámica no sea preocupante, pueden ser aplazadas. La presurización

permite que las reparaciones se puedan programar en jornadas normales de trabajo y en épocas favorables.

Las presurización contribuye a la reducción de los costos de mantenimiento y de los gastos de mano de obra en la atención al sistema (Jorge, 2011).

1.2.1. Limitaciones.

Mantener un cable presurizado no evita que se produzcan averías en los circuitos como consecuencia de daños importantes en las cubiertas de los cables.

Los cables y accesorios que se presurizan deben resistir la presión interna máxima prevista. Cuando por desperfectos la presión existente dentro de una coraza presurizada cae por debajo de la existente en el exterior se revierte el proceso y se absorbe el agua.

Para la aplicación de este sistema, en la red no pueden existir empalmes no presurizables o bloqueados con parafina u otro material que impida la libre circulación del gas (Pascual and Bocalandro, 2009).

Su implementación debe ser complementada con métodos apropiados de monitoreo que incluyan la medición precisa de la presión y flujo de gas y que permitan localizar y reparar las fugas antes que el servicio pueda ser afectado.

La distancia entre el punto de inyección y el punto más alejado de la red debe limitarse, de modo que se garantice la protección a todo lo largo del cable y en caso de fuga la presión no descienda por debajo de lo previsto sin que se accione el dispositivo de alarma (Bocalandro, 2005).

Se debe velar porque los equipos de deshidratación de aire no bajen su rendimiento, o el aire inyectado tendría un alto contenido de humedad (Domínguez, 2010).

1.3. Componentes de un sistema de presurización.

Un sistema de presurización (Figura 1.2) aplicado en una central telefónica, está formado por componentes internos y externos. Los componentes internos, ubicados en la central, forman parte del sistema de suministro de gas (aire); entre ellos se encuentran una fuente de alimentación de gas seco, un panel de distribución con sus medidores de flujo de gas, las tuberías de conexión y el sistema de alimentación al cable, integrado por válvulas y

conectores. Los componentes externos, se encuentran fuera de la central; son accesorios de presurización que forman parte del sistema; entre ellos se encuentran las mangas termocontraíbles, mangas mecánicas y los tapones (Jorge, 2011).

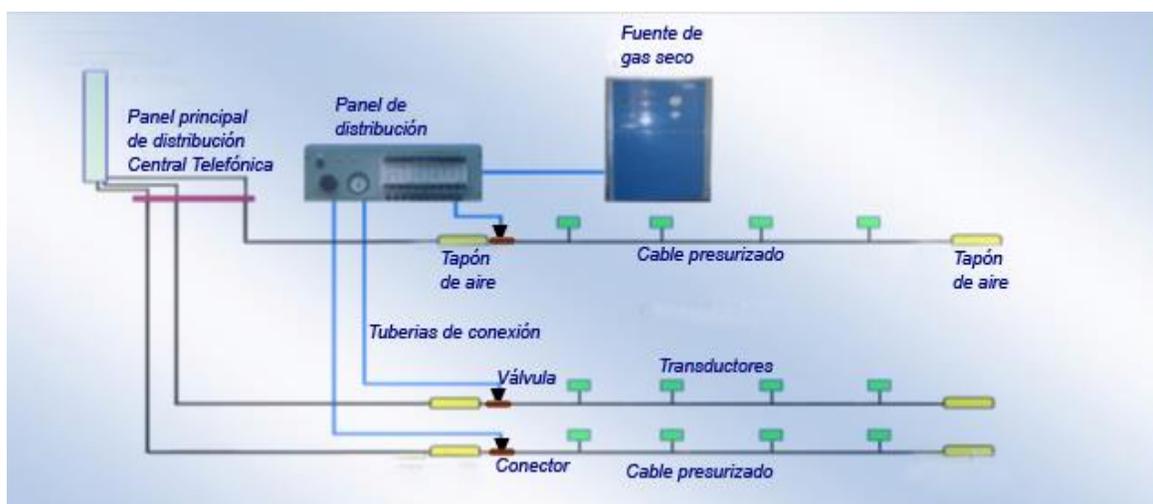


Figura 1.2. Esquema general de un sistema de presurización.

1.3.1. Fuente de alimentación.

Existen tres métodos fundamentales para el suministro de gas seco a los cables presurizados de las redes de telecomunicaciones (Jorge, 2011):

- Método de alimentación estática: consiste en la inyección de gas seco comprimido (nitrógeno) proveniente de cilindros neumáticos.
- Método de flujo continuo: consiste en la inyección permanente y a presión constante de gas seco (aire).
- Método de flujo continuo por tuberías: consiste en la inyección permanente de gas seco (aire) por medio de tuberías instaladas paralelamente a los cables.

Para los dos métodos de alimentación de flujo continuo la fuente de alimentación de gas seco es un compresor, cuya función incluye filtrar, comprimir y extraer la humedad al aire del ambiente que será inyectado a los cables presurizados como gas seco (Figura 1.2).

El compresor debe ser capaz de sustituir el volumen de gas que haya escapado de la red de cable bajo presión, mantener los valores de presión especificada en todos los cables de la red, poder alimentar la red de cable aun cuando se realicen ampliaciones y suministrar la cantidad suficiente de gas a una sección cualquiera de la red, de forma tal, que en caso de ocurrir una fuga la presión garantice protección (Labrado, 2010).

El método de flujo continuo es el más utilizado, porque no necesita un alto grado de hermeticidad de las corazas, permite ignorar fugas pequeñas y puede mantener la presión de protección mínima requerida (Pascual and Bocalandro, 2009).

1.3.2. Panel de distribución.

El panel de distribución o panel de flujo (Figura 1.2) se utiliza para distribuir y monitorear el aire, procedente de la fuente de alimentación, a cada uno de los cables presurizados.

El panel cuenta con instrumentos básicos, que permiten analizar y determinar las condiciones de funcionamiento de la red presurizada: un contador de volumen, encargado de registrar el volumen total de gas suministrado por la fuente y un medidor de flujo o rotámetro, que mide el flujo de gas que se suministra a cada cable (Pascual and Bocalandro, 2009).

Existen modernos paneles de flujo con tecnología digital, de menores dimensiones y que pueden ser encuestados directamente por la red telefónica conmutada o a través de una conexión ethernet por los sistemas de monitoreo de la presurización (Puregas, 2010).

1.3.3. Otros componentes del sistema de presurización.

En la Figura 1.2 se aprecian otros componentes del sistema de presurización; como la válvula de alimentación, la tubería que conecta el panel de distribución a dicha válvula, los conectores que permiten acoplar las válvulas a las corazas de los cables presurizados y los tapones de aire que se emplean para evitar que el aire que se encuentra dentro de las corazas se escape.

1.4. Sistemas de monitoreo y gestión de la presurización.

Un sistema de presurización debe ser complementado con métodos apropiados de gestión que permitan monitorear el funcionamiento del sistema en tiempo real, conocer las mediciones precisas de la presión y el flujo de gas en los cables y detectar anomalías en las cubiertas de las corazas que pudieran ser fuente de fugas. Los sistemas de monitoreo y gestión de la presurización son los encargados de supervisar y controlar constantemente la red presurizada, también son denominados sistemas de supervisión por telemedición (Jorge, 2011).

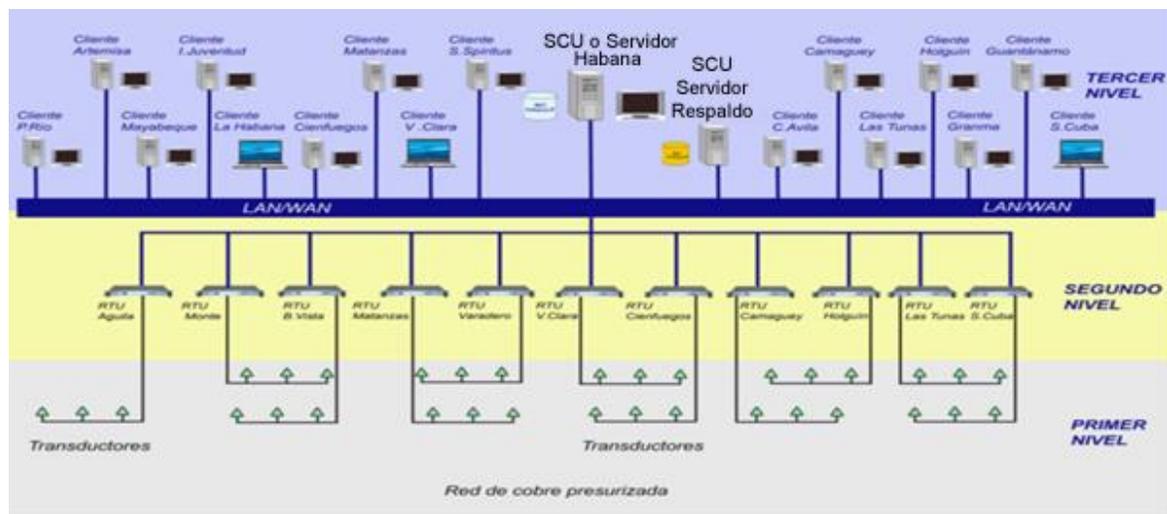


Figura 1.3. Topología de un Sistema de monitoreo y gestión de la presurización.

La inversión de cualquier tipo de sistema de monitorización tiene un valor inicial alto, pero sus resultados son sustanciales a mediano plazo. La realización del mantenimiento preventivo genera altos beneficios, evitando la sustitución de secciones enteras de cables y las pérdidas consecuentes a la interrupción del servicio. Estudios realizados demuestran que el costo de mantenimiento preventivo, como promedio, es de un tercio del costo de mantenimiento correctivo debido a los ahorros continuados (Jorge, 2011).

Estos sistemas basan su funcionamiento en tres niveles fundamentales (Pascual, 2007):

- Nivel 1 Transductores.
- Nivel 2 Unidades de Adquisición de Datos.
- Nivel 3 Unidad Central de Procesamiento.

1.4.1. Transductores.

Un transductor es un dispositivo al que se le aplica una energía de entrada y devuelve una energía de salida; esta energía de salida suele ser diferente al tipo de energía de entrada. Debido a la facilidad con la que se transmite y amplifica la energía eléctrica, los transductores más utilizados son los que convierten otras formas de energía como calor, luz o sonido; en energía eléctrica (Creus, 2010).

Los transductores de presión (TP) son dispositivos electrónicos que transforman la magnitud física de presión en una magnitud eléctrica. Para el monitoreo de las redes presurizadas de telecomunicaciones los transductores de presión más utilizados son aquellos que convierten la magnitud de presión en una señal de corriente eléctrica modulada por un tono de audio, cuya frecuencia tiene una determinada relación de conversión Presión/ Frecuencia, por ejemplo: 1 mbar = 1 Hz. Esta selección se debe, en lo fundamental, a las características de las señales que normalmente se transmiten por estas redes de telecomunicaciones y a las distancias a que son instalados los transductores.

Los transductores son instalados convenientemente en registros, lugares de fácil acceso, en empalmes donde existan cambios de calibre, derivaciones o ramificaciones del cable telefónico, en el comienzo y terminación de los cables. Se colocan a distancias que garanticen el cálculo del punto de fuga (Yumar and Martínez, 2014).

1.4.2. Unidades de adquisición de datos.

La unidad de adquisición de datos (RTU) del sistema de gestión de la presurización alimenta e interroga los transductores utilizando una rutina cíclica y recibe sus respuestas en frecuencia, almacena y controla los valores recibidos comparándolos con umbrales programados, verificando si existe alguna condición de alarma. Si esta condición se manifiesta, la unidad efectúa una transmisión de hacia la unidad central de procesamiento, además interactúa con esta para recepcionar los parámetros programados en ella (Puregas, 2002).

1.4.3. Unidad Central de Procesamiento.

Es una estación de trabajo con un software capaz de realizar las funciones de programación y visualización de la arquitectura de la red de cables presurizados, interroga periódicamente

o por pedido del operador a la unidad de adquisición asignada, para mantener un conocimiento permanente del estado del sistema y verificar su correcto funcionamiento, localiza puntos de fuga utilizando gráficas y almacena eventos para análisis estadísticos (NICOTRA, 2009).

En las versiones más avanzadas de monitorización de la presurización existen cuatro niveles, de manera que el tercer nivel se encarga de la supervisión regional y el cuarto nivel de la supervisión de los niveles regionales, de la redundancia de datos, de la carga de trabajo distribuida y del respaldo ante el fallo de una unidad de procesamiento del tercer nivel (Yumar and Martínez, 2014).

1.5. Sistemas de monitoreo y gestión de la presurización en el mundo.

Entre los sistemas de monitoreo de la presurización más difundidos en el mundo se encuentran:

- El Sistema de Monitoreo Unificado UMS (*Unified Monitoring System*) de Lancier, producido y comercializado por NATELCO S.A.C.I. es utilizado por Deutsche Telecom, Telecom Argentina y Telefónica de Brasil (Natelco, 2007), (Pascual, 2007), (Jorge, 2011), (Lancier, 2007).
- La solución de monitorización de la presurización de cables CPMS (*Cable Pressure Monitoring Solution*) desarrollado por Monitronix Europa. Algunos de los clientes que utilizan las soluciones de Monitronix son: Telstra, Cable & Gíreles Communications, Telecom, Batelco, Tella, TELIKOM PNG, Eircom y China Telecom. (Jorge, 2011), (Europe-Monitronix, 2011).
- Sistema de Supervisión de Cables Presurizados producido y comercializado por INELCOM. Es un software privado de Telefónica de España y se encuentra en explotación en ese país y sus Filiales desde 1988 (INELCOM, 2005), (Jorge, 2011), (Pascual, 2007).
- El Pressure MAP (*Management Analysis Program*), Programa de Análisis de Gestión de la Presurización desarrollado por la compañía privada norteamericana *System Studies Incorporated* (Jorge, 2011). El sistema se encuentra instalado en las principales operadoras telefónicas de Estados Unidos y en varias compañías de telecomunicaciones entre las que podemos mencionar: BellCanada, SaskTel Telia

(Suiza), Korea Telecom, Columbia Ministry of Communication (Kuwait), Elisa (Finlandia), entre otras (Airtalk, 2011).

- El Sistema PVMS de su sigla en inglés *Pure View Management System* desplegado por la firma Canadiense Puregas Equipment Limited (Puregas, 2010).
- El sistema de adquisición de datos NiDA2 (*Nicotra Data Acquisition*) fabricado por la firma Italiana Nicotra Sistemi spa y comercializado por Canadian Puregas Equipment Ltd. se encuentra instalado en numerosas compañías telefónicas de varios países, entre ellos: Brasil, Canadá, China, Alemania, Italia, Israel, Portugal, Uruguay, Sudáfrica y Cuba (Plath, 2007), (NICOTRA, 2002), (NICOTRA ,2011), (Jorge, 2011), (Labrado, 2010), (Pascual, 2007).

1.6. Estado de la presurización en ETECSA.

Teniendo en cuenta la importancia que reviste la actividad de presurización, ETECSA dirige sus esfuerzos a lograr su implementación en todas las corazas factibles de protección. No obstante en la actualidad no se alcanzan las cifras previstas. De un total de 1228 corazas instaladas, se encuentran presurizadas 1033 para un 84.1%.

El sistema de monitoreo supervisa 633 corazas para un 51.5% del total y un 61 % del total de corazas presurizadas.

En Camagüey, existen un total de 60 corazas instaladas, de ellas 56 ya se encuentran presurizadas para un 93.3%. La supervisión se extiende a 53 corazas para un 88.3% del total y un 94.6% de las corazas presurizadas. La situación de Camagüey se encuentra por encima de la media nacional pero el estado de la presurización a nivel nacional, no tiene un comportamiento similar ni tan favorable (ETECSA, 2015).

1.6.1. Estado de los sistemas de monitorización y gestión de la presurización en ETECSA.

Como se señala con anterioridad el sistema de adquisición de datos instalado en ETECSA es el NIDA 2, fabricado por la firma italiana Nicotra Sistemi spa, y comercializado por *Canadian Puregas Equipment Limited*, compañía canadiense (NICOTRA, 2002), (Puregas, 2002).

La topología implementada del sistema NiDA 2 es la configuración Cliente Servidor escalonadas en tres niveles (Figura 1.3).

En el primer nivel se encuentran las corazas presurizadas de los Centros de Telecomunicaciones (CT) con los transductores de presión y flujo. En ETECSA se encuentran instalados los transductores TP204M std de presión (ver Anexo 1).

El segundo nivel lo conforman las RTU (*Remote Terminal Unit*) o unidades de adquisición remotas. Actualmente las unidades utilizadas son OmniRACK. Esta representa una unidad avanzada de adquisición diseñada por Nicotra Sistemi spa para satisfacer las necesidades de las operadoras de telecomunicaciones en el control de la presión en cables de cobre y los sistemas de seguridad de estos. Su función es adquirir, de forma automática o bajo petición, los datos de los transductores de presión instalados y transferir los resultados a la Unidad de supervisión y control ubicada en el nivel 3 (Nicotra, 2012). Entre las principales características de este equipamiento se puede mencionar:

- Medición de los transductores de presión utilizando el módulo OmniPRESS.
- Gestión de transductores de seguridad con el módulo OmniACCESS8.
- Gestión del antirrobo de cables a través del módulo OmniCABLE25.

También cuenta con el módulo OmniCONTROL que incluye la sección de alimentación, la unidad de procesamiento central y la sección de comunicación del OmniRACK. Este módulo se basa en una versión integrada del sistema operativo Linux y dispone de un disco interno para almacenar aplicaciones y datos.

Cada módulo OmniPRESS tiene la posibilidad de gestionar hasta 16 líneas de adquisición de datos de transductores. En cada canal del módulo OmniPRESS se puede instalar un máximo de 127 transductores. Por lo tanto, la capacidad máxima de este módulo es de 2032 puntos de medición de la presión en la red presurizada.

La comunicación con la SCU (*Supervisor Control Unit*) o servidor del sistema NiDA 2 es a través de la conexión ethernet LAN /WAN o de una línea telefónica (NICOTRA, 2012).

Los criterios de modularidad y estructura común del OmniRACK proporcionan una elevada integrabilidad y uso compartido de los recursos entre diferentes sistemas. Sus ventajas

técnicas permiten al usuario interesado utilizar soluciones efectivas y generalizadas, para sus problemas de monitorización y gestión de la red.

A pesar de las ventajas y flexibilidad de las unidades de adquisición OmniRACK dista mucho de explotarse todas sus capacidades porque son equipos costosos y no se ha podido implementar todos los módulos funcionales por cuestiones financieras. Por esta misma causa la cantidad de unidades instaladas es pequeña. Teniendo que recurrir a soluciones de explotación que están por debajo de las posibilidades reales que tienen las unidades.

En el tercer nivel se ubica la supervisión y gestión centralizada que abarca el servidor, su respaldo y los clientes, ambos componentes son considerados unidades de control y supervisión (*SCU*), aunque con características distintivas. Estas unidades con una distribución equitativa de la técnica a supervisar accederían de forma remota a la base datos en el servidor, en este nivel se recibirían las mediciones y las alarmas de las RTU de los diferentes CT que componen la red, se ejecutarían las funciones de programación y visualización de la arquitectura del sistema, localización de puntos de fuga y almacenamiento de eventos para análisis estadísticos entre otras (NICOTRA, 2012). La aplicación del sistema de Adquisición de Datos NIDA2 en ETECSA, ha hecho posible que la empresa cuente con una herramienta de monitoreo y gestión de la presurización de muchas prestaciones.

A pesar de sus ventajas, el sistema de supervisión presenta también un conjunto de limitantes asociado fundamentalmente a los costos. El equipamiento y los dispositivos necesarios para la implementación del primer y segundo nivel (transductores y unidades de adquisición), que también son los más numerosos; tienen un costo elevado (Yumar and Martínez, 2014):

Unidad OmniRACK:	\$3 090.00
Transductor TP204M std:	\$190.25
Comprobador de transductores direccionables TPM-12:	\$3 500.00

En muchos casos, estos costos constituyen un freno importante para la implementación del sistema de monitoreo y gestión de la presurización sobre todo en aquellas regiones pequeñas donde no es factible económicamente la instalación de una unidad de adquisición.

Esto impide la monitorización de corazas presurizadas instaladas en pequeñas localidades o ubicaciones remotas y obliga a implementar esquemas que no permiten explotar a cabalidad las capacidades del sistema.

Otra limitación, asociada a patentes y licencias, se relaciona con la procedencia del software de gestión que es privativo, como se señaló anteriormente. Es propiedad de la firma italiana *Nicotra Sistemi spa* y comercializado por *Canadian Puregas Equipment Ltd*, compañía canadiense. Esta situación impide la adaptación del sistema a las necesidades particulares de nuestro entorno ya que carece de la flexibilidad necesaria o el acceso a su diseño para modificarlo de acuerdo a nuestros intereses.

1.7. Consideraciones finales del capítulo.

La presurización es de vital importancia para garantizar el mantenimiento de los parámetros eléctricos de la red telefónica de cables de cobre. Gran parte de la calidad de los servicios brindados a través de las mismas dependen de estas características. Complementar la presurización con métodos apropiados de monitoreo y gestión permite un uso más eficiente de las capacidades instaladas, reduce las interrupciones y el gasto de recursos necesarios para su eliminación.

La Empresa de Telecomunicaciones de Cuba cuenta con un sistema de gestión y monitoreo de red de cables de cobre presurizada. Los altos costos de los componentes de este sistema (hardware y software) impiden que el mismo se explote a cabalidad. El monitoreo de la red no alcanza a todos los cables presurizados en el territorio.

Esta problemática ilustra la necesidad real de desarrollar soluciones alternativas y de costos reducidos, para lograr un desarrollo sostenible de esta actividad.

CAPÍTULO 2. Implementación de la unidad de adquisición de datos.

Para el desarrollo de la unidad de adquisición de datos se realiza un grupo de tareas, teniendo como objetivo principal la medición de la frecuencia de los tonos de audio generados por los transductores de presión. Las características técnicas de los mismos así como la naturaleza y temporización de sus señales son descritas en este capítulo. La plataforma de desarrollo Arduino, seleccionada para la implementación de esta solución, es también analizada; exponiéndose las principales características del hardware y del software utilizado para su programación. Se abordan temas referentes a la interface necesaria para acondicionar la señal y los algoritmos para realizar la función definida como objetivo principal, en este caso medir la frecuencia emitida por cada uno de los transductores de presión instalados en una coraza presurizada.

El capítulo consta de siete epígrafes. Se hace referencia a los datos técnicos de los transductores de presión. Se abordan los métodos de medición de frecuencia, el método seleccionado para realizar este trabajo y sus principales ventajas. Se exponen las razones por las que se escoge la plataforma Arduino en general para desarrollar esta investigación, detallando las características técnicas, aplicaciones, herramientas y ventajas del proyecto Arduino. Del modelo Arduino Ethernet R3, en específico, se detallan las características técnicas que determinaron su selección para la implementación de la unidad de adquisición; objetivo general de esta investigación. Se muestra la inserción de la unidad de adquisición de datos implementada, en el sistema de gestión y monitoreo de la presurización en ETECSA, además las características generales de la interface necesaria para acondicionar la señal, recibida desde transductores, de acuerdo a los requerimientos de entrada de la unidad de adquisición implementada. Se tratan los algoritmos para medir frecuencia y para medir las señales emitidas por los transductores de presión. Las conclusiones parciales de este capítulo son expuestas en el último epígrafe.

2.1. Características técnicas y funcionamiento de los transductores de presión TP204M std.

Los transductores de presión son los elementos primarios del sistema en los cuales se centra este estudio. Conocer las características del funcionamiento de los transductores y de las

señales que generan los mismos es imprescindible para desarrollar los algoritmos y la interfaz necesaria para el tratamiento de estas señales en la unidad de adquisición de datos.

Los transductores de presión utilizados en ETECSA son del modelo TP204M (std), diseñados para el monitoreo de cables presurizados. Estos transmiten a la unidad de adquisición, un tono cuya frecuencia es función del valor absoluto de la presión medida, a través del mismo par telefónico por el que reciben el voltaje de alimentación. Su direccionamiento debe ser definido antes de la instalación, por medio de jumpers, permitiendo la identificación de hasta 127 transductores conectados en la misma línea (Anexo 1) (Yumar and Martínez, 2014), (Benítez, 2010).



Figura 2.1. Circuito interno del transductor TP204M (std).

La figura 2.1, muestra el circuito interno de un transductor TP204M (std) y los pines de conexión para colocar los jumpers de direccionamiento.

Los pines se encuentran numerados como se puede observar en la figura 2.2a. El direccionado de se logra cada transductor a través de combinaciones de jumpers siguiendo la estructura de un código binario (Anexo 1). Por ejemplo, para programar la dirección número 12 se cortocircuitan los pines marcados con los números 4 y 8 (Figura 2.2b).

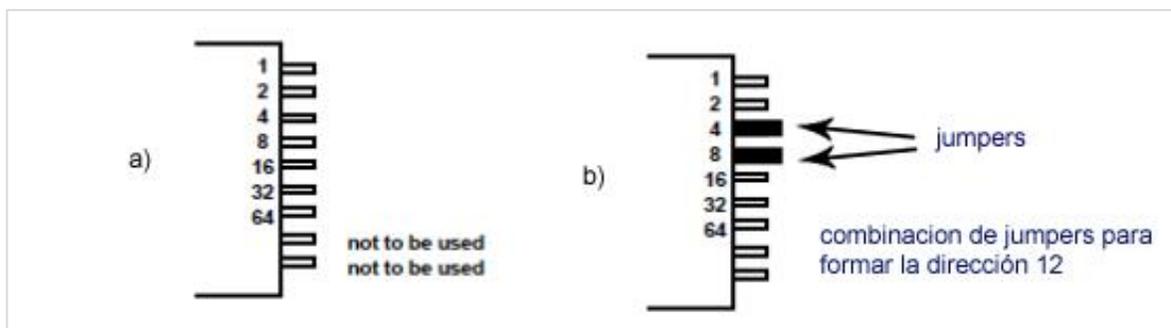


Figura 2.2. Diagrama y ejemplo de direccionado del TP204M (std).

Cuando se energiza la línea telefónica a la que está conectado un transductor, con el voltage requerido; después de transcurrido un período de tiempo (expresado en segundos), determinado por la dirección configurada en el transductor multiplicada por dos; el mismo emite una señal de corriente modulada durante 1.5 segundos. Esta señal es proporcional a 1 Hz por mBar de presión existente en la coraza. Por lo general las presiones mantenidas dentro de las coraza se encuentran entre 1000 y 2000 mBar. Los transductores TP204M (std) tienen un rango de medición de 800 a 2000 mBar.

Durante este período inicial de espera el consumo de corriente del transductor es entre 25 y 50 μA , durante el tiempo de trabajo el transductor consume típicamente una corriente de 5.5 mA.

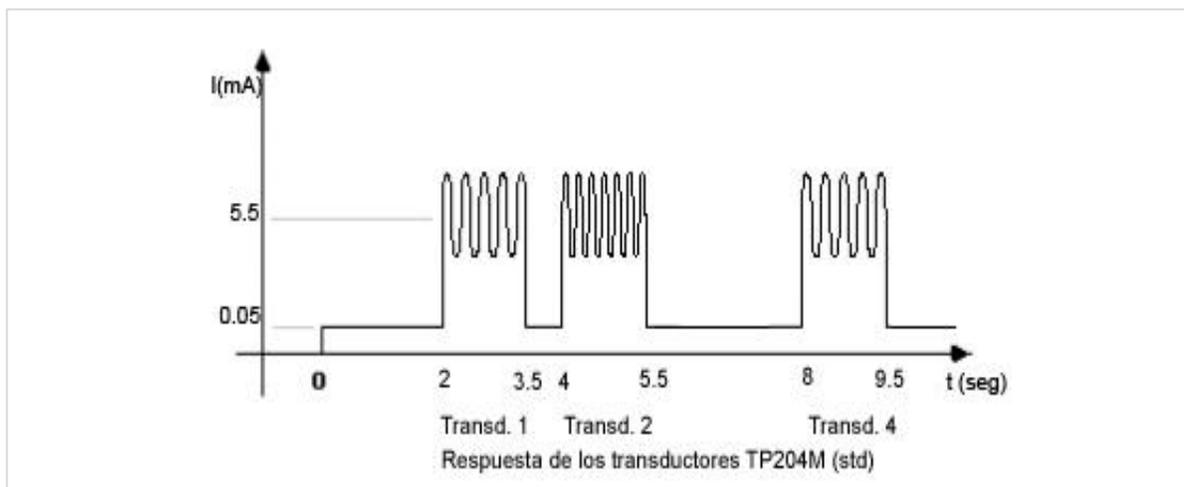


Figura 2.3. Tiempo de respuesta de tres transductores TP204M (std).

La figura 2.3 muestra el tiempo de respuesta para el caso particular de tres transductores. Las direcciones de los transductores son 1, 2 y 4. Como puede apreciarse el tiempo de espera inicial de los transductores 1 y 2 es de 2 y 4 segundos respectivamente. En el caso del transductor 4 su tiempo de espera inicial es de 8 segundos. Estos tiempos están en correspondencia con las direcciones configuradas en los dispositivos.

2.2. Métodos de medición de frecuencia.

Un contador digital convencional es un dispositivo electrónico que mide la frecuencia de una señal de entrada, como medida indirecta del resultado de la cuenta directa de eventos. En su diseño también se incluyen las opciones de medida de períodos, cocientes entre las frecuencias de las señales aplicadas a sus entradas, medidas de intervalos temporales entre dos eventos y funcionamiento en modo totalizador.

El esquema básico de un contador con sus principales elementos se presenta en la figura 2.4. Este diagrama en bloque muestra la disposición del instrumento en el modo de funcionamiento como frecuencímetro (González, 2015), (Packard, 2015).

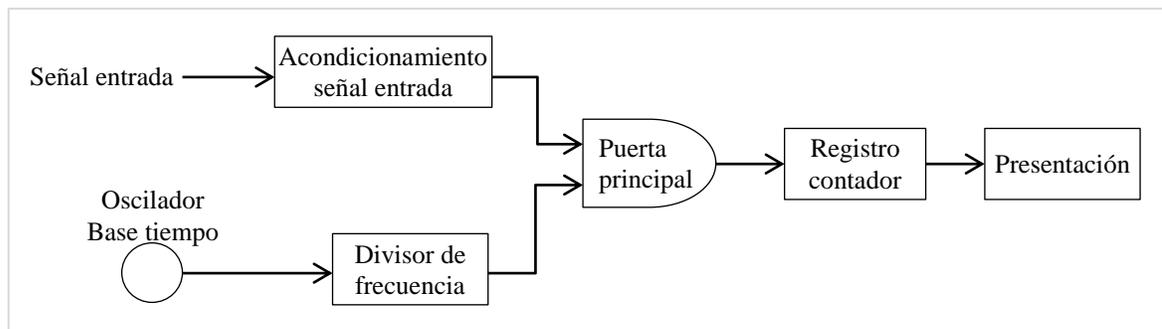


Figura 2.4. Esquema básico de un contador convencional.

La señal de entrada se acondiciona inicialmente con el fin de hacerla compatible con los circuitos del contador. El resultado del acondicionamiento es un tren de pulsos, cada uno de los cuales representa un ciclo de la señal original. Con la puerta principal abierta, los pulsos pasan al registro-contador. La base de tiempos controla el tiempo de apertura de la puerta; este tiempo suele denominarse también tiempo de medida; y es el tiempo que dura la realización del conteo.

La precisión de la medida depende de la precisión con que se determine este tiempo; es decir, el error cometido en la medida depende de la exactitud de la base de tiempos. En consecuencia, la mayoría de los contadores emplean osciladores de cristal con frecuencias de 1, 5 ó 10 MHz como elementos básicos de temporización en sus bases de tiempos. El divisor de la base de tiempos toma esta frecuencia y la divide en intervalos, que constituyen los tiempos de puerta en el modo frecuencímetro y los pulsos a contar en el modo periodómetro (González, 2015), (Packard, 2015).

2.2.1. Contadores recíprocos.

Este conjunto de instrumentos realizan siempre una medida del periodo de la señal como punto de partida para posteriores cálculos (casi todos los contadores operan según este principio). Si la magnitud de interés es la frecuencia, ésta se calcula como la inversa del período y se presenta de forma directa (Figura 2.5).

Esta técnica se ha extendido en el diseño de contadores debido a dos características principales (González, 2015), (Packard, 2015):

- El error de cuantificación o error de cuenta ± 1 es independiente de la frecuencia de la señal de entrada (en modo periodómetro depende sólo de la frecuencia de la base de tiempos). Por consiguiente, la resolución del instrumento es independiente de la frecuencia de la señal de entrada.
- Se puede controlar la puerta principal en tiempo real. A este rasgo se le denomina capacidad de “rearme” de la puerta.

Con el fin de apreciar las ventajas de este método basta considerar los errores asociados a la indecisión digital, el error de cuenta ± 1 en los modos frecuencímetro y periodómetro. En el modo de funcionamiento frecuencímetro, el error cometido es el inverso de la frecuencia de la señal de entrada:

Error relativo en modo frecuencímetro $\frac{\Delta f}{f} = \pm \frac{1}{fe}$, donde fe es la frecuencia de la señal de entrada.

Mientras que en el modo periodómetro este error relativo está definido por la siguiente expresión:

Error relativo modo periodómetro $\frac{\Delta T}{T} = \pm \frac{Tc}{\max(Te, tp)}$, donde Tc es el período de base de tiempo y $\max(Te, tp)$ es el máximo entre el período de la señal de entrada (Te) y el período de la señal que controla la puerta (tp).

En el modo periodómetro, si se selecciona un tiempo de puerta mayor que el periodo de la señal de entrada, el instrumento toma el tiempo de puerta seleccionado. Sin embargo, si el periodo de la señal de entrada es mayor que el tiempo de puerta, dará tiempo a que se produzca un nuevo rearme.

El diagrama de bloques básico de un contador recíproco es similar al de un contador convencional salvo por el detalle de que incorpora dos registros contadores en lugar de uno. Un registro está destinado a la cuenta de ciclos de reloj y el otro está destinado a la cuenta de los impulsos generados por la señal de entrada. Ambos contadores contabilizan impulsos durante el mismo tiempo de puerta. La figura 2.5 muestra un sencillo diagrama de bloques en el que se aprecian los elementos anteriores (González, 2015), (Packard, 2015).

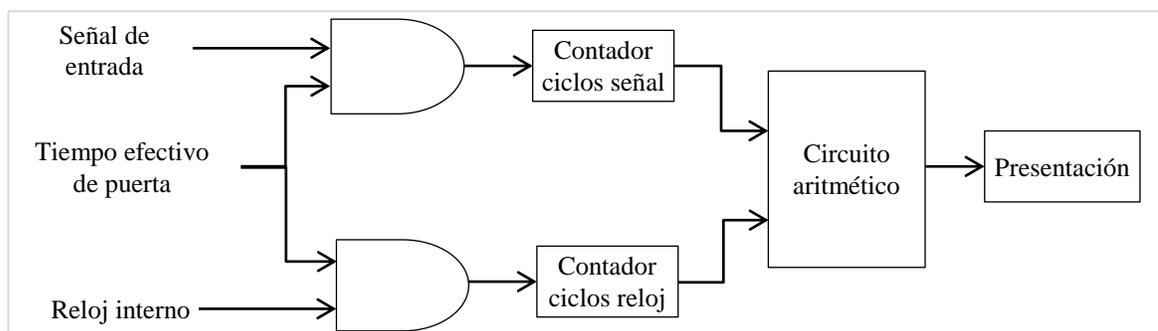


Figura 2.5. Esquema básico de un contador recíproco.

Este es el esquema básico del método de medición de frecuencia seleccionado para el desarrollo de los algoritmos necesarios en la implementación de unidad de adquisición de datos.

2.3. Plataforma Arduino.

Arduino, es la plataforma escogida para la implementación de la unidad adquisición de datos. Se escogió Arduino, debido a que es una plataforma de desarrollo de prototipos con electrónica abierta, con herramientas de programación basadas en software libre y es fácilmente programable desde una PC. Arduino es un producto de bajo costo del cual existe variada documentación, multitud de herramientas y librerías de funciones de licencias libres.

Arduino, consiste en una placa con un microcontrolador, el cual se programa desde el ordenador mediante el software Arduino, conectando la placa al ordenador por USB. El lenguaje de programación utilizado es propio, y está basado en *Wiring*, que a su vez está basado en *Processing*, y la sintaxis es igual que la de C++. Está pensado para que sea

flexible y fácil de usar (no necesariamente con conocimientos avanzados de electrónica o programación) y está teniendo una amplia difusión, debido a la gran cantidad de tutoriales y proyectos desarrollados y documentados en la web, a su bajo coste y facilidad de uso (Torrente, 2015).

2.3.1. Origen del proyecto Arduino.

Arduino nació en el año 2005 en el Instituto de Diseño Interactivo de Ivrea (Italia), centro académico donde los estudiantes se dedicaban a experimentar con la interacción entre humanos y diferentes dispositivos (muchos de ellos basados en microcontroladores) para conseguir generar espacios únicos, especialmente artísticos. Arduino aparece por la necesidad de contar con un dispositivo para utilizar en las aulas que fuera de bajo coste, que funcionase bajo cualquier sistema operativo y que contase con documentación adaptada a gente que quisiera empezar de cero. La idea original fue, fabricar la placa para uso interno de la escuela. No obstante, el Instituto se vio obligado a cerrar sus puertas precisamente en 2005. Ante la perspectiva de perder en el olvido todo el desarrollo del proyecto Arduino, se decide liberarlo y abrirlo a “la comunidad”. La colaboración de muchas personas ha hecho que Arduino poco a poco haya llegado a ser lo que es actualmente: un proyecto de hardware y software libre de ámbito mundial (Torrente, 2013).

2.3.2. Fundamentos del proyecto Arduino.

El proyecto Arduino está constituido por tres elementos fundamentales:

- Una placa hardware libre. La placa Arduino es hardware libre porque sus ficheros esquemáticos están disponibles para descargar de la página web del proyecto con la licencia “*Creative Commons*”, la cual es una licencia libre que permite realizar trabajos derivados tanto personales como comerciales (siempre que estos trabajos den crédito al proyecto Arduino y publiquen sus diseños bajo la misma licencia). En particular, el hardware libre permite que las interesadas puedan estudiarlo para entender su funcionamiento, modificarlo, reutilizarlo, mejorarlo y compartir dichos cambios (Torrente, 2013).
- Un entorno de desarrollo gratis, libre y multiplataforma. El entorno de desarrollo integrado (en inglés *integrated development environment*, IDE) Arduino se publica

con una combinación de la licencia GPL (para el entorno visual de programación propiamente dicho y la licencia LGPL (para los códigos fuente de gestión y control del microcontrolador a nivel más interno). La consecuencia de esto es, que cualquier persona, puede formar parte del desarrollo del software Arduino y contribuir así a mejorar dicho software, aportando nuevas características, sugiriendo ideas de nuevas, funcionalidades, compartiendo soluciones a posibles errores existentes, etc. Es multiplataforma porque se puede instalar y ejecutar en sistemas operativos Windows, Mac OS X y Linux (Torrente, 2013), (Arduino, 2014).

- El lenguaje Arduino (distribuido bajo las mismas licencias del IDE) basado internamente en código C/C++. El lenguaje puede ser extendido a través de librerías de C++, los especialistas y técnicos pueden cambiar al lenguaje de programación AVR C en el que está este basado. AVR C es un subconjunto del lenguaje C,



Figura 2.6. Elementos del proyecto Arduino. diseñado específicamente para los microcontroladores AVR de Atmel. El conjunto de instrucciones AVR está implementado físicamente y disponible en el mercado en diferentes modelos de microcontroladores, que comparten el mismo núcleo AVR pero tienen distintos periféricos y cantidades de RAM y ROM (Arduino, 2014).

2.3.3. El microcontrolador y la placa

La placa Arduino, está centrada en los microcontroladores del fabricante Atmel. Aunque en la actualidad existen modelos basados en microcontroladores de otros fabricantes. Además incluye una interfaz USB, que utiliza para comunicarse con una computadora y obtener alimentación eléctrica. La función de la placa es permitir la programación del microcontrolador, mediante conexión USB o ICSP (*In Circuit Serial Programming*), y proporcionar un fácil acceso a todos sus terminales, mediante los pines laterales de la placa. A través de estos terminales puede o bien leer señales del exterior, o generar señales de control.

Existen varias placas Arduino oficiales, cada una con diferentes características como el tamaño físico, el número de pines ofrecidos, el modelo de microcontrolador incorporado y como consecuencia, entre otras cosas, la cantidad de memoria utilizable (Torrente, 2013).

Un microcontrolador es un sistema basado en un procesador, que está construido en un solo chip, y que puede usarse para desempeñar tareas muy diversas. Los modelos usados en la placa básica Arduino son en esencia iguales, y, simplificando mucho, consisten en (Torrente, 2015):

- Un procesador
- Una memoria flash en la que se almacena el programa que ejecuta el procesador
- Una memoria SRAM donde se guardan las variables del programa
- Una memoria EEPROM, para almacenar datos que perduren al apagar el sistema
- Puertos de entrada/salida, conectados directamente a los terminales del chip
- Módulos de comunicaciones, para transferencia de datos según distintos protocolos: UART (*Universal Asynchronous Receiver Transmitter*), SPI (*Serial Peripheral Interface*) y TWI (*Two Wire Interface*, más frecuentemente llamado I2C, (*Inter-Integrated Circuit*)).
- Módulos contadores, para funciones de temporización
- Módulos para modulación por anchura de pulso
- Conversores analógico/digital

Los puertos de entrada/salida ocupan todos los pines del chip, salvo los de alimentación y tierra. Cada pin se puede configurar como entrada, permitiendo al procesador leer el valor digital de la tensión en ese pin, o como salida, para que el procesador pueda controlar el nivel de tensión del pin escribiéndole un valor digital (0 para 0V, 1 para 5V).

El resto de los módulos nombrados para funcionar también pueden necesitar utilizar algunos pines del chip, así que, cuando estén en funcionamiento, esos pines no podrán usarse como entrada/salida (además hay un pin que no suele usarse, pues tiene función de reset).

2.3.4. Expansiones o shields

Además de las placas Arduino propiamente dichas, también existen los llamados “*shields*” (en español, “escudo”) o expansiones. Una expansión no es más que una placa de circuito impreso que se coloca en la parte superior de una placa Arduino y se conecta a ella mediante el acoplamiento de sus pines sin necesidad de ningún cable. Su función es actuar como placas suplementarias, ampliando las capacidades y complementando la funcionalidad de la placa base Arduino de una forma compacta y estable. Así un proyecto hecho en la plataforma Arduino suele consistir en una placa base, una o varias expansiones y un software (*sketch*) de Arduino que las controla (Torrente, 2013).

2.4. Arduino Ethernet R3.

Para seleccionar el modelo de la plataforma de desarrollo Arduino, se analizan y valoran las características y funcionalidades de la unidad de adquisición de datos OmniRACK, que se exponen en el epígrafe 1.6.1. Los principales aspectos valorados son: la adquisición de datos de transductores conectados a una línea telefónica, el procesamiento y almacenamiento de los datos obtenidos y la comunicación con la SCU o servidor del sistema de monitoreo y gestión de la presurización, que se realiza mediante una conexión ethernet. También se toma en cuenta la posibilidad de configuración remota de los dispositivos.

El Arduino Ethernet R3 es una placa basada en el microcontrolador ATmega328. La misma cuenta con 14 entradas/salidas digitales, seis entradas analógicas, un oscilador a 16 MHz controlado por un cristal o un resonador de cuarzo, una interfaz ethernet con un conector

RJ45, un conector para la fuente de alimentación, una entrada ICSP, un lector de tarjetas microSD y un botón de reset (Arduino, 2014), (Torrente, 2013). Las características de este modelo de Arduino, potenciadas por los periféricos integrados en su microcontrolador o incluidos en la placa base, brindan grandes posibilidades para implementar una unidad de adquisición de datos que realice funciones similares a las unidades instaladas actualmente.

2.4.1. Características técnicas generales:

Microcontrolador	ATmega328
Voltaje de operación	5V
Conexión de voltaje de entrada (recomendado)	7-12V
Conexión de voltaje de entrada (límites)	6-20V
Voltaje de entrada PoE (límites)	36-57V
Pines E/S Digitales	14 (de los cuales 4 proveen salida PWM)
Pines entradas analógicas	6
Corriente CD por pin E/S	40 mA
Corriente CD para pin 3.3V	50 mA
Memoria Flash	32 KB (ATmega328) de ellos 0.5 KB usados por el cargador
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad de reloj	16 MHz
Controlador ethernet integrado	W5100 TCP/IP
Lector de tarjeta Micro SD,	con convertidor activo de voltaje.

2.4.2. Entradas y salidas.

La placa Arduino Ethernet R3 posee catorce pines digitales que pueden ser usados como entradas o salidas. Estos pines funcionan con un nivel de voltaje de 5 Volt. Cada pin proporciona o recibe un máximo de 40 mA y tiene una resistencia interna “pull-up”, desconectada por defecto, de un valor entre 20 y 50 Kohm. Adicionalmente algunos pines tienen funciones especializadas (Torrente, 2015):

-
- Comunicación serie: pines 0 y 1, usados para recibir (RX) y transmitir (TX) datos serie respectivamente.
 - Interrupciones externas: pines 2 y 3. Estos pines pueden ser configurados para disparar una interrupción en un nivel bajo, por flanco de subida, bajada o por cambio de nivel.
 - Modulación de Ancho de Pulso (PWM, por sus siglas en inglés): pines 3, 5, 6, 9 y 10, proporcionan una salida a una frecuencia predeterminada con modulación de ancho de pulso, permitiendo definir el ancho de los pulsos generados con una resolución de 8 bit, es decir, 256 valores.
 - SPI: pines 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK), estos pines soportan la comunicación SPI (*Serial Program Interface*).
 - LED: Pin 9. Este pin está conectado a un diodo emisor de luz (LED) incluido en la placa. Cuando el pin 9 tiene un valor alto (5 Volt), el LED está encendido, cuando tiene un nivel bajo, el LED está apagado.

Las entradas analógicas, nombradas desde A0 hasta A5, proporcionan una resolución de 10 bits (1024 niveles) cada una. Por defecto, el rango de medición es de 0 a 5 Volt, pero es posible cambiar el límite superior utilizando el pin AREF (Torrente, 2015).

También algunos pines tienen funciones especializadas:

- TWI: A4 (SDA) y A5 (SCL). Soportan la comunicación I²C (*Two Wire Interface*).

El pin Reset: llevando esta línea a un nivel bajo, se puede resetear el microcontrolador. Típicamente es utilizado a par añadir un botón de reset a las expansiones (Torrente, 2015).

2.4.3. Comunicaciones.

El Arduino Ethernet posee un número de facilidades para comunicarse con una computadora, otro Arduino u otros microcontroladores.

La librería SoftwareSerial permite la comunicación serie con otros Arduinos utilizando los pines digitales.

El Atmega 328 también soporta comunicación I²C (*Two Wire Interface*) y SPI. El software Arduino también incluye la librería WIRE para simplificar el uso del bus I²C (*Two Wire Interface*), para la comunicación SPI (*Serial Program Interface*) se utiliza la librería SPI.

La placa también puede ser conectada a red Ethernet. La librería Ethernet y otras similares permiten implementar diversos servicios y funciones (Torrente, 2015).

2.5. Inserción de la unidad de adquisición en el Sistema de gestión y monitoreo de ETECSA.

La unidad de adquisición de datos alternativa se ubica dentro del esquema del sistema de monitoreo y gestión de la red de cables presurizada de ETECSA en el segundo nivel (Figura 2.7). Su utilización será una alternativa para poder incluir en el sistema, aquellas

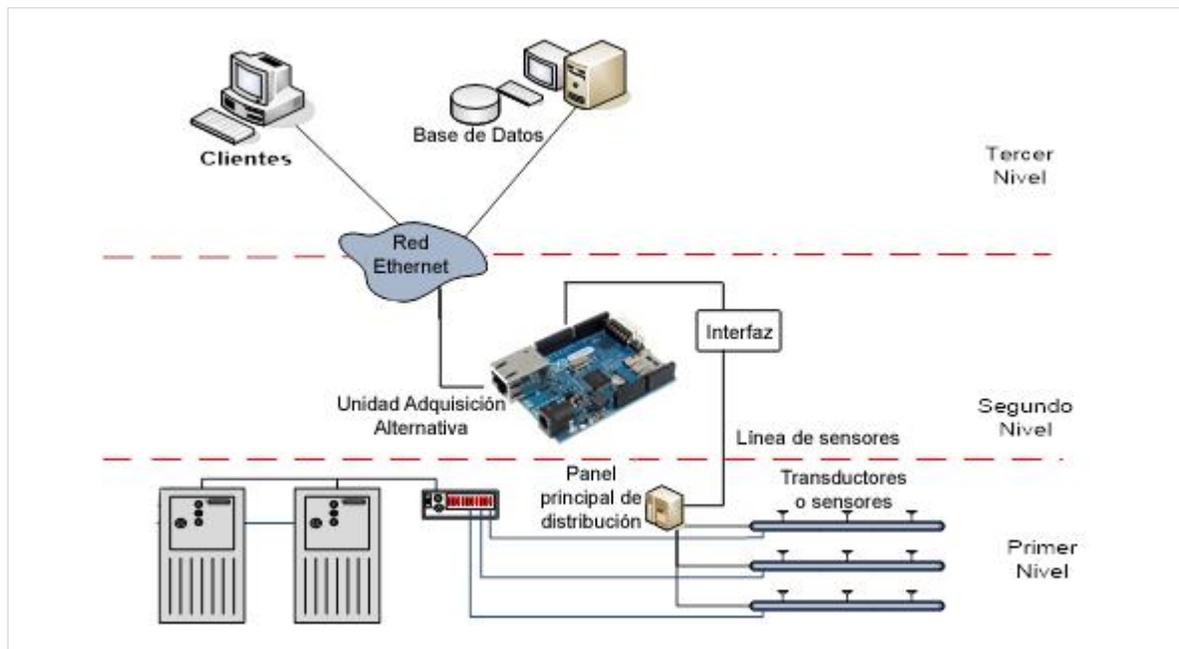


Figura 2.7. Ubicación de la unidad de adquisición en el sistema de monitoreo y gestión de la red presurizada de ETECSA.

localidades donde por las características de la red presurizada, por los elevados costos u otras causas, no sea posible utilizar equipos profesionales. Su integración al sistema será similar a la de las unidades de adquisición OmniRACK, que se utilizan en ETECSA.

En la figura 2.7 se muestra, entre la línea de transductores y la unidad de adquisición una interfaz, que fue implementada en la Dirección Territorial de ETECSA en Camagüey, en trabajos que antecedieron a esta investigación (Yumar and Martínez, 2014). Esta interfaz

suministra el voltaje necesario para el funcionamiento de los transductores en cada ciclo de medición, acondiciona la señal emitida por los transductores de presión, de manera que la misma sea compatible con la señal de entrada procesada por la unidad de adquisición.

La integración de la unidad de adquisición alternativa al sistema de monitoreo y gestión de la presurización, se realiza a través de la red ethernet; para lo cual el hardware seleccionado cuenta con la interfaz necesaria. Este tema no se aborda en el presente proyecto, porque se propone su desarrollo en futuras investigaciones.

2.6. Algoritmo para medir frecuencia.

Para la medición de frecuencia de las señales emitidas por los transductores de presión se utiliza la librería `FreqMeassure` del proyecto Arduino. La misma implementa, apoyándose en el hardware del microcontrolador, un contador recíproco.

2.7. Algoritmo para la medición de las señales emitidas por los transductores.

El algoritmo desarrollado para la medición de las señales emitidas por los transductores de presión se muestra en el Anexo 2. Su principal característica es la ejecución repetitiva del proceso de obtención de datos según una planificación preestablecida en la configuración de la unidad de adquisición, es decir, el proceso de medición se ejecuta cada determinado tiempo.

El algoritmo está dividido en tres bloques funcionales: “Inicialización”, “Atención a interrupciones” y “Adquisición de datos”. El primer bloque, “Inicialización”, comienza en el conector 1 y se ejecuta una vez. Es el encargado de la inicialización de las variables de control del algoritmo, obteniendo sus valores de los datos de configuración de la unidad de adquisición o mediante un proceso de cálculo.

El bloque “Atención a interrupciones” comienza en el conector 2; se ejecuta cada un segundo, disparado por la interrupción 0 (`int0`) generada por hardware. La señal de interrupción proviene de una placa de expansión de reloj de tiempo real (RTC, *Real Time Clock* por sus siglas en inglés). La función de este bloque es mantener actualizadas las variables de control de tiempo y las variables de control del bloque “Adquisición de datos”. Estas últimas variables controlan el momento de inicio o finalización de cada ciclo de medición, de acuerdo a los datos de configuración de la unidad de adquisición. Además,

este bloque del algoritmo es el encargado de emitir la señal a la interfaz para conectar y desconectar la alimentación a la línea de transductores.

El bloque “Adquisición de datos” que comienza en el conector 3, se ejecuta continuamente, pero la función de adquisición de datos sólo comienza cuando las variables de control del ciclo toman los valores adecuados en el bloque anterior. Este bloque es el encargado, además de captar las señales de los transductores de presión, procesarlas y enviarlas a través de una conexión serie a una computadora.

2.8. Conclusiones parciales.

En este capítulo se realiza un estudio de las características y del funcionamiento de los transductores de presión TP204M (std), profundizándose en las señales generadas por los mismos. Se seleccionó el contador recíproco como método de medición de las señales emitidas por los transductores. Se investiga sobre las características, funciones y ventajas del proyecto Arduino, los distintos modelos existentes, las placas de expansiones y otros aspectos de interés. Esta investigación permite seleccionar el modelo Arduino Ethernet R3, como el más adecuado para implementar la unidad de adquisición de datos porque cuenta entre sus características: entradas y salidas analógicas y digitales, interfaces serie UART y ethernet para comunicaciones, y un lector de tarjetas microSD para el almacenamiento de datos.

En el capítulo además se crearon los algoritmos para la medición de frecuencias y para la medición de las señales emitidas por los transductores conectados a la línea presurizada de ETECSA.

CAPÍTULO 3. Resultados

En este capítulo se describen los experimentos y simulaciones realizados para comprobar la precisión y el funcionamiento de los algoritmos diseñados e implementados para la unidad de adquisición de datos. Los datos obtenidos de los experimentos son procesados estadísticamente para determinar la exactitud y la incertidumbre de los valores. Las simulaciones se realizaron utilizando el software Proteus versión 7.7 SP2, con el objetivo de validar el correcto funcionamiento del algoritmo de acuerdo a las necesidades planteadas.

El capítulo está estructurado en 3 epígrafes. Se comenta sobre la medición de frecuencia con Arduino, y sobre los resultados obtenidos, sobre la validación del algoritmo de medición para una línea de transductores de presión TP204M (std) mediante simulación por computadora incluyendo la simulación de la línea de transductores y de la unidad de adquisición de datos; además se realiza una valoración económica y medioambiental que ilustra el impacto de este trabajo.

3.1. Medición de frecuencia con Arduino.

Con el objetivo de comprobar la calidad de la placa Arduino como medidor de frecuencia se implementa, utilizando la librería FreqMeasure del proyecto Arduino, un software para medir frecuencia y registrar las mediciones en una computadora para su posterior análisis (Anexo 3).

Se realiza un experimento para comprobar la exactitud con la cual se puede medir frecuencia con una placa Arduino Ethernet R3. El experimento consiste en comparar las mediciones tomadas con la placa Arduino con las mediciones realizadas con un frecuencímetro (Figura 3.1).

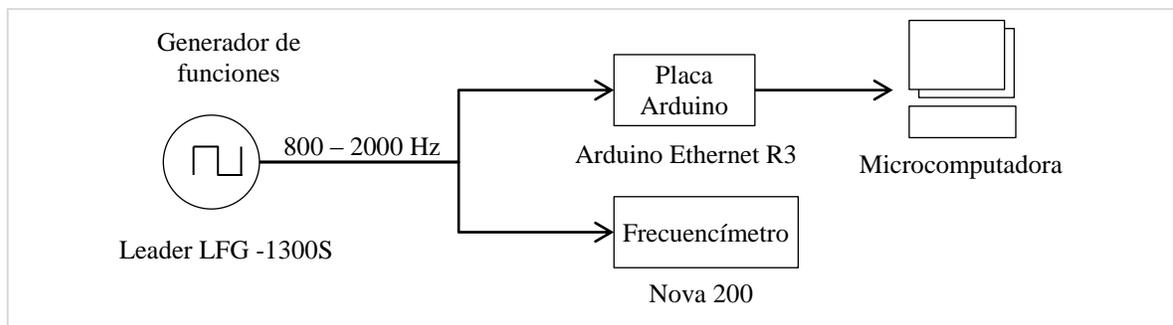


Figura 3.1. Esquema del experimento para medir frecuencia con Arduino.

Una señal de onda cuadrada con niveles TTL, obtenida de un generador de funciones Leader LFG-1300S (Anexo 4), se inyecta simultáneamente a un frecuencímetro Nova 200 (Anexo 5) y a la placa Arduino (Figura 3.2). Se anota la lectura obtenida en el frecuencímetro y las mediciones realizadas por la placa Arduino son transmitidas a una computadora a través de una conexión serie y almacenadas para su posterior análisis. Las mediciones se realizan en todo el rango de trabajo de la unidad de adquisición de datos (800 a 2000 Hz) a intervalos de 200 Hz. Durante la prueba la temperatura es de 24°C.

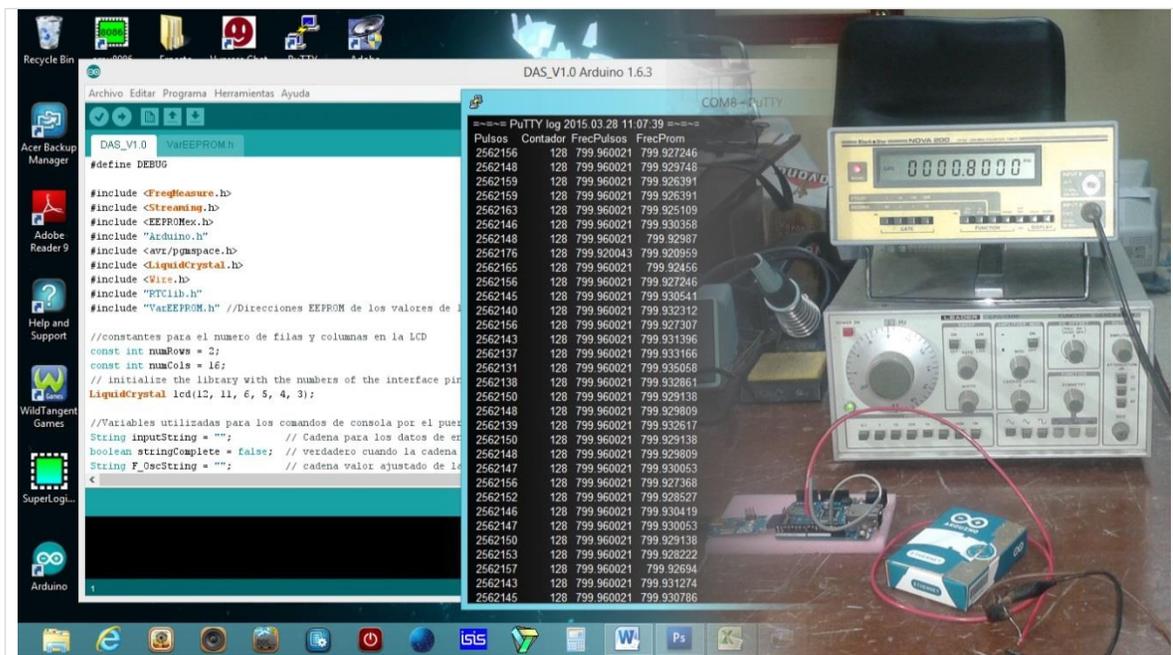


Figura 3.2. Fotos del experimento para medir frecuencia con Arduino.

3.1.1. Resultados obtenidos en la medición de frecuencia.

Durante la realización del experimento descrito en el epígrafe anterior, los resultados de las mediciones fueron transmitidos a una microcomputadora, a través de una conexión serie. Los datos fueron almacenados y procesados posteriormente.

El resultado de una medición está completo únicamente cuando está acompañado de una declaración cuantitativa de la incertidumbre, que expresa la calidad del mismo y permite valorar la confiabilidad de este resultado (Sáez and Font, 2001).

Para valorar la viabilidad de utilizar el Arduino Ethernet R3 en la medición de frecuencia, se determina la incertidumbre total de las mediciones realizadas, que está compuesta por la

incertidumbre combinada de las mediciones y la incertidumbre del patrón de referencia utilizado (Centro Español de Metrología, 2008), (Sáez and Font, 2001).

Para calcular la incertidumbre de las mediciones se utiliza el software Microsoft Excel por la facilidad que ofrece el mismo en el trabajo con grandes volúmenes de datos tabulados.

Las tablas con una síntesis de los valores obtenidos en las mediciones realizadas y los cálculos de la incertidumbre para los siete valores de frecuencia seleccionados, pueden ser consultadas en el Anexo 6.

En el anexo 7, tabla 8, se pueden consultar los valores de incertidumbre de las diferentes fuentes asociadas al patrón de referencia. También se puede observar la incertidumbre total para cada valor de frecuencia seleccionado en el intervalo de 800 a 2000 Hz. Las principales fuentes de incertidumbre del contador de frecuencia Nova 200 (Anexo 4) son: la estabilidad de la base de tiempo, estabilidad con cambios de temperatura, envejecimiento, resolución y error de cuenta ± 1 .

Los resultados finales de las mediciones para cada una de las frecuencias medidas con la placa Arduino Ethernet R3, así como su incertidumbre expandida se muestran en la tabla 3.1.

Tabla 3.1. Resultados de las mediciones y su incertidumbre.

Frecuencia Patrón (Hz)	Incertidumbre del Patrón*	Frecuencia Medida (Hz)	Número de Mediciones	Incertidumbre Mediciones	Incertidumbre combinada	Incertidumbre expandida
800	± 0.030	799.94	3530	± 0.001	± 0.030	± 0.061
1000	± 0.031	999.96	4416	± 0.001	± 0.031	± 0.062
1200	± 0.032	1200.39	2889	± 0.002	± 0.032	± 0.064
1400	± 0.033	1400.13	2412	± 0.003	± 0.033	± 0.066
1600	± 0.034	1600.08	1455	± 0.002	± 0.034	± 0.068
1800	± 0.035	1800.11	2855	± 0.002	± 0.035	± 0.071
2000	± 0.037	2000.07	15561	± 0.001	± 0.037	± 0.073

* Incertidumbre combinada de las principales fuentes de incertidumbre del patrón de frecuencia.

La incertidumbre expandida (Tabla 3.1) se calcula utilizando un coeficiente de Student $k=2$ para una confiabilidad del 95.45% de las mediciones (Centro Español de

Metrología, 2008). Como se observa en dicha tabla el mayor valor de incertidumbre obtenido es de ± 0.073 por lo que se puede afirmar que en todo el rango de medición el Arduino Ethernet R3 no tendrá un error de medición superior a este valor con una confiabilidad del 95.45%.

Este resultado final certifica que Arduino Ethernet R3 se puede utilizar para la medición de frecuencia en la implementación de la unidad de adquisición de datos porque la precisión requerida para en esta función no es mayor de 1 Hz, según se deduce de los estudios realizados anteriores a esta investigación (Yumar and Martínez 2014).

3.2. Implementación del software para la unidad de adquisición de datos.

Basado en el algoritmo diseñado con anterioridad, en el epígrafe 2.7, se implementa el software de medición de una línea de transductores de presión TP204M (std). Esta tarea se realiza utilizando las herramientas de programación del proyecto Arduino (IDE Arduino y librerías del proyecto).

El software configura y controla distintos periféricos que forman parte de la unidad de adquisición entre los que se encuentran: un reloj de tiempo real (RTC), una pantalla de cristal líquido (LCD) opcional, un controlador de comunicación serie (UART) y el acceso a la memoria (EEPROM), donde se graban las variables de configuración de la unidad de adquisición. El código obtenido se expone en el Anexo 8.

3.3. Validación del algoritmo de medición para una línea de transductores de presión TP204M (std) mediante simulación por computadora.

Para la validación del funcionamiento correcto del algoritmo de medición de una línea de transductores de presión se realiza una simulación computacional. Se simula el circuito electrónico de la unidad de adquisición de datos, que incluye un módulo virtual de la placa Arduino (Sain, 2014). El software de simulación empleado para esta tarea es el Proteus versión 7.7 Service Pack 2 (Sharma, 2014).

El módulo virtual de simulación de la placa Arduino corresponde al modelo “Arduino UNO”, cuya única diferencia con el modelo Arduino Ethernet R3 es que no incluye el módulo de interfaz de comunicación ethernet, sin embargo, en todos las demás características son similares.

Esta simulación incluye la generación de señales equivalentes a las generadas por los transductores utilizados en ETECSA.

3.3.1. Simulación de la línea de transductores.

Para comprobar el funcionamiento del algoritmo, es necesario en primer lugar realizar la simulación del comportamiento de una línea de transductores. Partiendo del estudio realizado de los transductores TP204M (std) tratados con anterioridad en el epígrafe 2.1 se logra obtener las señales equivalentes al comportamiento de dichos transductores, en este caso se simulan ocho transductores con direcciones no continuas. Esta simulación es encapsulada en un bloque funcional o sub-circuito nombrado en la simulación “LINE SENSOR SIMULATION” (Anexo 9).

3.3.2. Simulación de la unidad de adquisición de datos.

Se implementa en Proteus los circuitos equivalentes de la unidad de adquisición de datos utilizando un módulo virtual para la simulación de la placa Arduino (ver ANEXO 9). Se incluye en la simulación otros componentes electrónicos que pueden ser adicionados a una unidad de adquisición real. En la imagen mostrada en el Anexo 9 se puede apreciar el bloque funcional correspondiente a la línea de transductores tratada en el epígrafe anterior; un circuito equivalente a una expansión de reloj de tiempo real, del proyecto Arduino (DS1307), la cual es utilizada para suministrar a la unidad de adquisición de datos la fecha y la hora actualizadas, necesarias para la sincronización de los ciclos de medición. Además se puede observar una pantalla LCD que permite la visualización de informaciones del estado de la unidad así como los resultados de las operaciones. También se incluye un terminal virtual que simula en Proteus un terminal serie de computadora.

Mediante esta simulación se puede comprobar el correcto funcionamiento del algoritmo diseñado y la interconexión y programación de los diferentes componentes de la unidad de adquisición de datos (reloj de tiempo real, memoria de almacenamiento de datos, presentación de resultados, transmisión de datos por un puerto serie, configuración y calibración del dispositivo).

3.4. Valoración económica y medioambiental.

La aplicación del sistema de Adquisición de Datos NiDA versión 2 en ETECSA, ha hecho posible que la empresa disfrute de las ventajas que propicia la aplicación de un sistema de monitoreo y gestión de la presurización. Pero los costos elevados de los componentes que integran el sistema dificultan su adecuada implementación y explotación.

La generalización de una solución como la desarrollada en este trabajo, permite como se plantea en los epígrafes iniciales, extender el monitoreo y gestión de la red presurizada a todos los territorios que aún no cuentan con estas funciones; además del costo reducido, se lograría un considerable ahorro de recursos materiales, humanos y de tiempo. Se evitarían la realización de grandes reparaciones y por consiguiente se reduciría el impacto en el medio ambiente.

El coste monetario estimado para el desarrollo de este diseño es de 100 CUC aproximadamente. Este valor representa sólo el 3% del costo de los equipos que actualmente están en uso en ETECSA.

3.5. Conclusiones parciales del capítulo

Como resultado del proceso de investigación, en este capítulo se llega a las siguientes conclusiones:

- Se diseña un sistema alternativo para la adquisición de los valores de presión de las corazas presurizadas.
- La medición de frecuencia se realiza con una incertidumbre de un valor de centésimas de Hertz, lo que corrobora la calidad de las mediciones y califican los resultados del sistema como confiables.
- Se comprueba mediante simulación que el algoritmo implementado funciona adecuadamente.
- Se implementa el software para la medición de frecuencia en la unidad de adquisición de datos.

Conclusiones

Del análisis de los resultados logrados se arriba a las siguientes conclusiones:

- La presurización de la red de cobre representa una tarea importante de protección al evitar el deterioro en los cables, conservando un correcto nivel de aislamiento entre los pares que conforman las corazas. La necesidad de un sistema de monitorización para controlar el estado de la presurización en tiempo real es algo imprescindible.
- El sistema de monitoreo y gestión de la red de cables de cobre presurizada de ETECSA requiere de un equipamiento que tienen un elevado costo de adquisición y mantenimiento además cubre solamente el 84 % de toda la red presurizada y el 51.5% solamente es supervisado en todo el país.
- La plataforma Arduino, por sus características, es adecuada para el desarrollo de sistemas de medición, monitoreo y control de amplio uso en los sistemas de comunicaciones.
- El algoritmo desarrollado, permite programar el software empleado para la medición de los tonos de frecuencia obtenidos de los transductores de presión. Estos logran una precisión y confiabilidad que se ajustan a las necesidades del problema.
- El diseño propuesto basado en la plataforma Arduino capta los valores de frecuencia emitidos por los transductores de presión.

Recomendaciones

Teniendo en cuenta los resultados alcanzados, se realizan las siguientes recomendaciones:

- Continuar el desarrollo de herramientas de hardware y software que posibiliten el desarrollo automatizando de unidades de adquisición de datos interactuando con las novedosas tecnologías.
- Hacer extensivo el uso de esta unidad de adquisición de datos a otros territorios del país por la importancia que reviste y por la necesidad de su utilización.
- Implementar en el Arduino el software que permita la transmisión de datos al tercer nivel , para garantizar la comunicación entre todos los elementos de sistema de monitoreo y gestión de la red de cables de cobre presurizada de ETECSA.

Referencias bibliográficas

- AIRTALK. 2011. System Studies Incorporated Cable Pressure Technologies [Online]. [Accessed noviembre del 2014].
- Agilent Technologies. 2015. Fundamentals of the Electronic Counters. [Online] USA: Available: www.agilent.com. [Accessed enero del 2015].
- Arduino. 2014. Home Page. [Online] Italy. Available: www.arduino.cc [Accessed 20 de noviembre del 2014].
- Benítez Méndez, Nelson. 2012. Herramienta informática para localizar puntos de fuga en la red de cables de cobre presurizada. Universidad Central “Marta Abreu” de Las Villas.
- Belleza, E. & Szymancyck, O. 2000. Diseño de Planta Externa.
- Bocalandro, J. 2005. Básico de presurización. In: ETECSA, C. D. F. N. (ed.). La Habana.
- CREUS, A. 2010. Instrumentación Industrial. In: ALFAOMEGA, E. (ed.) 8va ed.
- Centro Español de Metrología. 2008. Evaluación de datos de medición — Guía para la expresión de la incertidumbre de medida.
- Domínguez, S. 2010. Problema de humedad en cables multipares [Online].
- ETECSA, 2015. Informe sobre el estado de la presurización en el mes de febrero del 2015.
- EUROPE- MONITRONIX. 2011. CPMS Cable Pressure Monitoring Solutions [Online].
- González de la Rosa, Juan Jose. 2014. Instrumentación Electrónica, [Online]
- Hewlett C Fundamentals of the electronic counters, AN 200, Electronic counter series, 2000.
- INELCOM. 2005. Sistema de Supervisión de Cables Presurizados [Online].
- Jorge, G. 2011. Monitorización y gestión de las redes presurizadas en ETECSA. Universidad Central “Marta Abreu” de Las Villas.
- Labrado, L. 2010. Sistemas de presurización en redes de cables de cobre de planta exterior. Universidad Central “Marta Abreu” de Las Villas.
- LANCIER. 2007. UMS Unified Monitoring System [Online].

-
- Murata Manufacturing Co. Ltd. 2014. Ceramic Resonators.
- NATELCO 2007. UMS Unified Monitoring System.
- NICOTRA SISTEMI spa. 2002. Nicotra Nida 2 System Pressurized Copper Cable Network Monitoring and Management System.
- NICOTRA SISTEMI spa. 2003. MINIDAS. User's Guide. Rev. 1.21.
- NICOTRA SISTEMI 2009. Introduction to Nicotra Sistemi spa NiDA2 System In: Puregas, C. (ed.). Canada.
- NICOTRA SISTEMI 2011. NiDA - Pressurized Copper Cable Network Monitoring System.
- NICOTRA SISTEMI. TP204M (std) Data Sheet.
- Óscar Torrente Artero. 2015. Curso práctico de formación.
- Pascual, R. 2007. Propuesta de un Sistema de Monitoreo y Gestión de la Red de Cobre Presurizada en ETECSA. Universidad Central “Marta Abreu” de Las Villas.
- Pascual, R. & Bocalandro, J. 2009. Integral de Presurización. In: ETECSA, C. N. D. C. (ed.). Santa Clara.
- Plath, I. 2006. Secadores de aire para las comunicaciones [Online]. Italia: canadian puregas equipment Available: www.canadianpuregas.com. [Accessed 13 de marzo 2012].
- Plath, I. 2007. Ni.DA 2xx. Arquitectura General.
- Puregas. 2002. Sistemas de Monitoreo de Presurización.
- Puregas. 2010. Pure View Information Management System.
- Sharma, Vikas. 2014. ARDUINO SIMULATION USING PROTEUS (ISIS) [Online] Available: 2embeddedrobotics.blogspot.com. [Accessed 7 de diciembre del 2014].
- Sáez Ruiz, S. J. & Font Ávila, Luis. 2001. Incertidumbre de la medición: Teoría y práctica. 1ra ed. Maracay. Estado Aragua. Venezuela. 2001.
- SEC-VAC. 2010. La calidad de transmisión en cables de comunicaciones [Online]. Argentina.

S. Wolf, y R.F.M. Smith, Guía para mediciones electrónicas y prácticas de laboratorio, edición ampliada y actualizada. Prentice-hall hispanoamericana. México, etc., 1992.

Torrente Artero, Oscar. 2013. Arduino. Curso práctico de formación. Alfaomega Grupo Editor, México, febrero 2013

W.D. Cooper and A.D. Helfrick, Instrumentación electrónica moderna y técnicas de medición, Prentice-Hall. Hispanoamericana, 1991.

Yumar Hernández, A. & Martínez Rosabal, O. 2014. Implementación del Sistema de Adquisición Alternativo para el Monitoreo en la Presurización.

Zain Nasi, S. 2014. Arduino Library For Proteus. A complete step by step tutorial on How to simulate Arduino in Proteus [Online] Available: <http://www.theengineeringprojects.com>. [Accessed 12 de diciembre del 2014].

Anexo 1

Datos técnicos del transductor de presión TP204M (std)

TP 204M (std)

DATA SHEET

PRESSURE TRANSDUCER



Description

The TP 204M (std) is a pressure transducer designed for the pressurized cables monitoring.

It transmits to the Acquisition Unit, a frequency proportional to the measured absolute pressure value, through the same pair from which it receives the power supply.

Its addressing is defined before the installation by jumpers, permitting to identify up to 127 transducers connected on the same line.

Also available in Long Distance version (LD) and Long Distance extended pressure range version (LDF).

Main characteristics

- Low power consumption
- Addressing capability
- Compensated in temperature
- High measurement accuracy
- Microprocessor driven
- Reduced size
- High reliability
- Linear response
- CE compliant

Specific characteristics

- Pressure range 800 ÷ 2000 mBar
- Frequency output 800 ÷ 2000 Hz
- Operating distance up to 40 Km

Address setting

- Remove the cap of the case containing the transducer;
- Set the jumpers on the available 7 pairs of pins as per the following *Addressing Table*, tacking into account the needs to univocally identify each transducer on the line, however remind to do not set an already used address;
- 2 pairs of pin are for testing purpose only (see figure 1);
- Adjust the cap properly and connect the wires to the acquisition unit dedicated pair line.

NICOTRA TRANSDUCER ADDRESSING TABLE																							
CODE	1	2	4	8	16	32	64	CODE	1	2	4	8	16	32	64	CODE	1	2	4	8	16	32	64
1								65								27							
2								66								28							
3								67								29							
4								68								30							
5								69								31							
6								70								32							
7								71								33							
8								72								34							
9								73								35							
10								74								36							
11								75								37							
12								76								38							
13								77								39							
14								78								40							
15								79								41							
16								80								42							
17								81								43							
18								82								44							
19								83								45							
20								84								46							
21								85								47							
22								86								48							
23								87								49							
24								88								50							
25								89								51							
26								90								52							
27								91								53							
28								92								54							
29								93								55							
30								94								56							
31								95								57							
32								96								58							
																59							
																60							
																61							
																62							
																63							
																64							

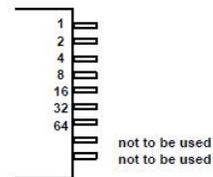


Figure 1

Installation

The TP204M (std) can be installed in two ways:

- A. "In Splice" inside the cable joints.
- B. "Out of Splice" using one of the MB series external waterproof cases.
(In this case the TP204M transducer wires are provided with faston connectors for the proper connection inside the MB enclosure)

Tools

- LC442/DP3: Portable Terminal and Test Unit performs: the line test, the single transducer test or the test of the whole transducers connected to a line.
- DM100M: Digital Pressure Gauge is used to measure the pressure value in the cable network at any testing points.

Programming

To program the TP204M (std) on Ni.D.A software please follow the instructions reported in MP8000 User's Guide.

Other related products

MB series: are manifolds designed to enclose the transducers when an Out of Splice installation is required. For further information please refer to the MB data sheets.

Related documentation

Pressure Transducer Installation	# 0062.E.
TP204M mounting procedure into MB 1/2 manifold	# 0010.E.
MP8000 User's Guide	# 0001.E.

Packing

The TP204M (std) are shipped in suitable boxes each ones containing: n°10 transducers, n°4 addressing jumpers each, the relevant data sheet and a silica-gel bag.

Technical characteristics

Supply Voltage :	25÷120 Vdc
Stand by current :	50 µA (typical), 25 µA (effective)
Working current :	5.5 mA (typical)
Frequency response range :	800 ÷ 2000 Hz
Signal level :	> 0 dBm
Addressing code per transducer line :	up to 127
Total measuring cycle time :	2 sec.
Pressure measurement range :	800 ÷ 2000 mBar
Relation air pressure / frequency :	1 mBar ≅ 1 Hz
Accuracy :	±0.5 % F.S. in the whole temperature range
Operating temperature range :	-20 ÷ +70 °C
Storage temperature range :	-40 ÷ +80 °C
Pulsed Voltage rejections :	1500 V - 1/1000 µsec, (better than CCITT K17)
Dimensions :	94 x 27 x 17 mm
Enclosure material :	Plastic
Signal wire :	22 AWG, 0.3 m
Weight :	35 gr.

Ordering Code

The TP 204M (std) without connectors on the wires is:	10000127
The TP 204M (std) with faston connectors is:	10000129

Specifications are subject to change without notice



Nicotra Sistemi spa

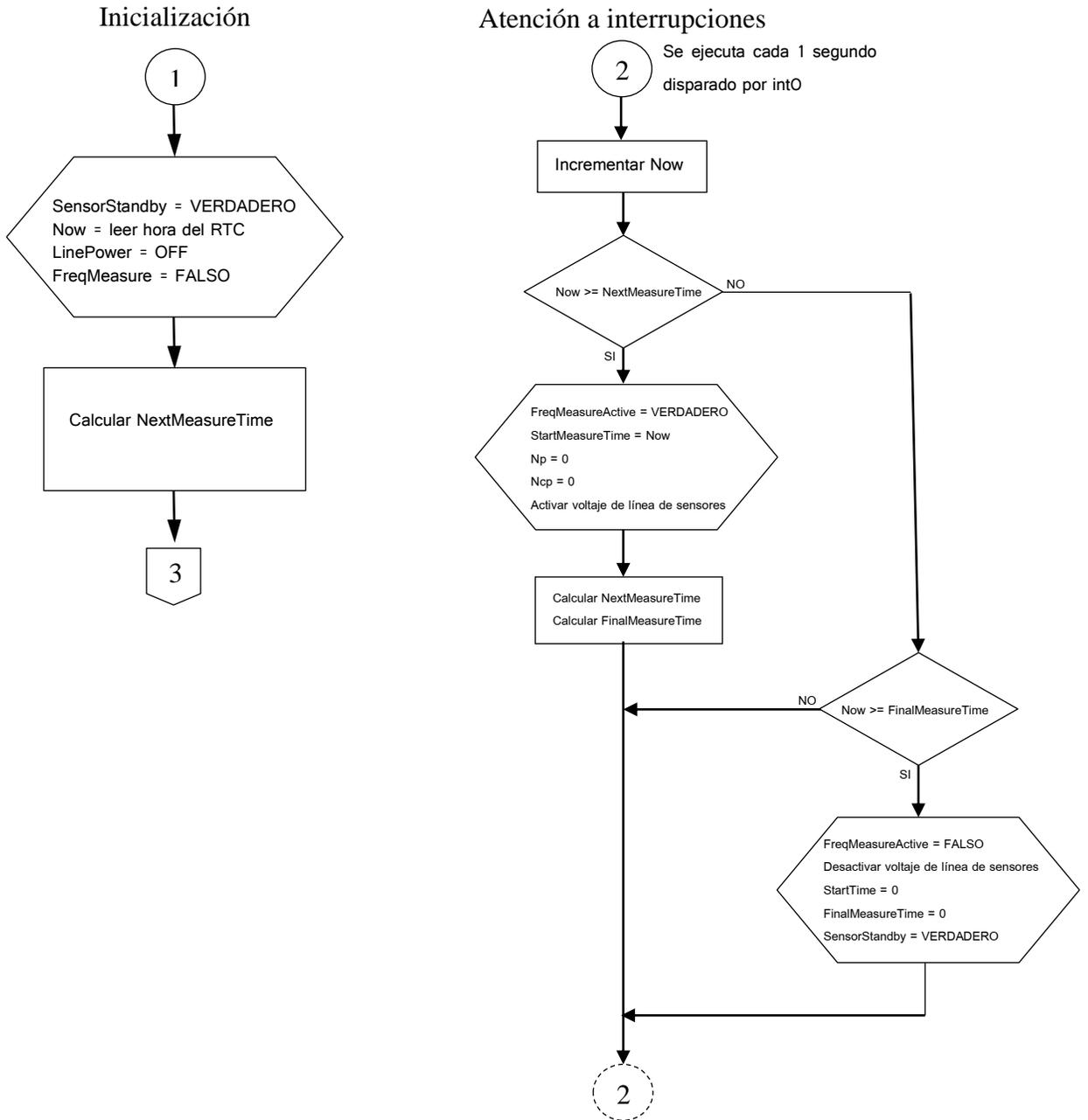
Via I Maggio 41 - 20021 Baranzate di Bollate - MI - Italy

Tel. +39 02 35689.1 - Fax. +39 02 35689.208 - E-mail: nicotra.sistemi@nicotragroup.com



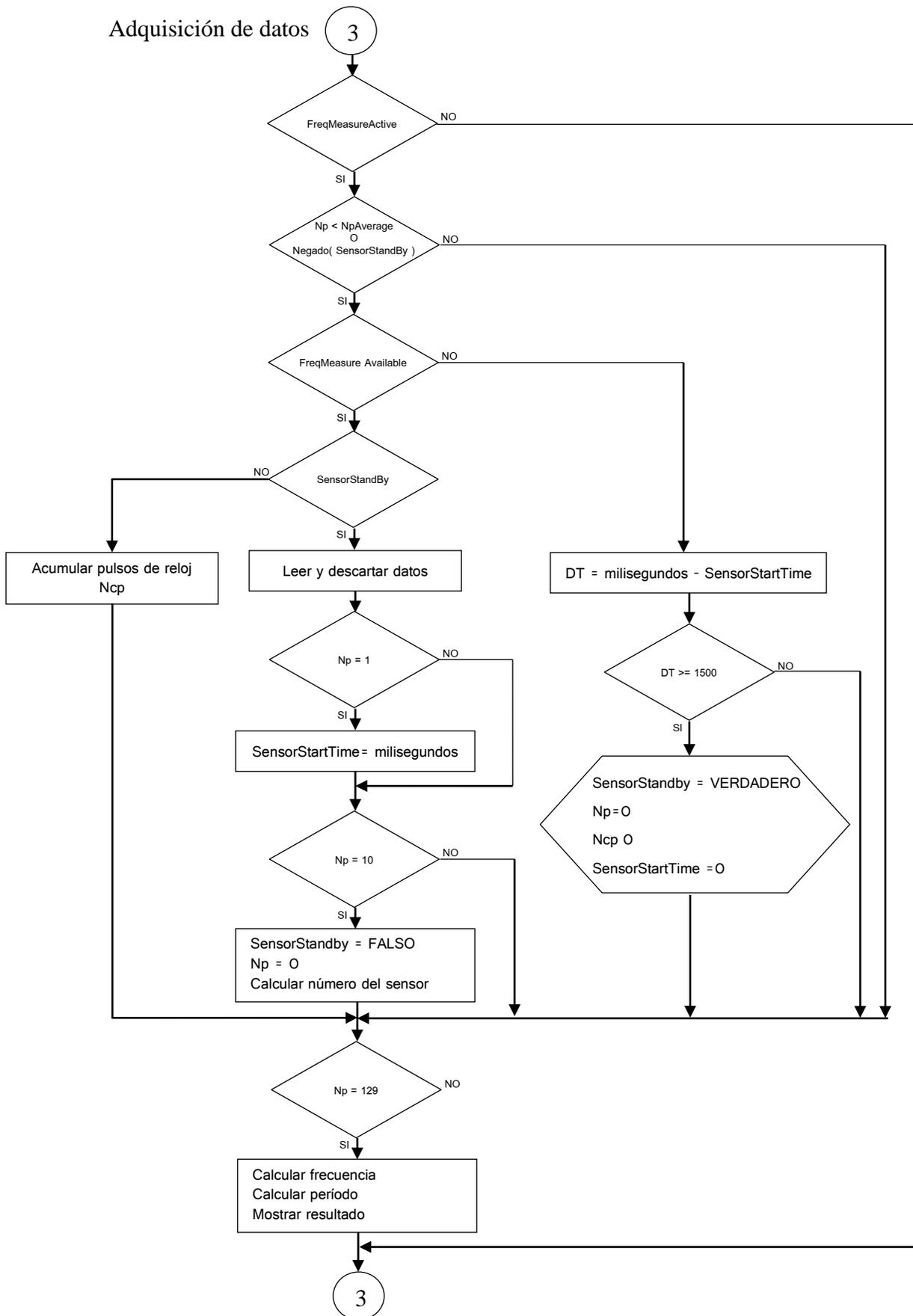
Anexo 2

Algoritmo para la medición de una línea de transductores TP204M (std) con la unidad de adquisición de datos.



Adquisición de datos

3



Anexo 3

Software para medir frecuencia y registrar mediciones con Arduino Ethernet R3

```
/******  
* Frecuencimetro con Arduino, utilizando valores de configuración de la unidad de adquisición  
*****  
#include <FreqMeasure.h>  
#include <Streaming.h>  
#include <EEPROMex.h>  
#include "Arduino.h"  
#include <avr/pgmspace.h>  
  
String inputString = ""; // Cadena para almacenar comando de entrada  
String F_OscString = "";  
String Np_AverageString = "";  
  
boolean stringComplete = false; // Cuando la cadena esta completa  
boolean FreqMeasureActive = true;  
unsigned long Ncp = 0; // Numero de pulsos del oscilador  
unsigned int Np = 0; // Numero de periodos (NP) de la señal de entrada  
float Period, Frequency;  
  
unsigned long F_Osc = 16000000;  
unsigned int Np_Average = 127; // Np_Average-1  
bool first = true;  
  
const int Np_Average_Addr PROGMEM = 54; // Numero de períodos a promediar  
const int F_Osc_Addr PROGMEM = 56; // Frecuencia ajustada del oscilador  
  
void setup() {  
  Serial.begin(115200);  
  Np_Average = EEPROM.readInt(pgm_read_byte(&Np_Average_Addr));  
  F_Osc = EEPROM.readLong(pgm_read_byte(&F_Osc_Addr));  
  
  Serial << "Medicion de Frecuencia y Periodo para " << F_Osc << " Mhz" << endl;  
  Serial.println("Np,Ncp,Periodo,Frecuencia");  
  inputString.reserve(25);  
  FreqMeasureActive = true;  
  FreqMeasure.begin();  
}  
  
void loop() {  
  // Imprime la cadena cuando se recibe un cambio de linea  
  if (stringComplete) {  
    if (inputString[0]=='\r'){  
      Serial.println("Puerto serie listo...");  
    }else{  
      inputString.toUpperCase();  
      if ((inputString.startsWith("OSC") || inputString.startsWith("OSC ")) && inputString.length()<=5){  
        Serial.print(String(F_Osc).substring(0,2) + "." + String(F_Osc).substring(2,11)); Serial.println("MHZ");  
      }else  
        if (inputString.startsWith("OSC ") && inputString.length()==13){
```

```

    F_OscString = inputString.substring(4,12);
    F_Osc = F_OscString.toInt();
    //Escribe el valor de la frecuencia del oscilador ajustada para la EEPROM
    EEPROM.writeLong(pgm_read_byte(&Np_Average_Addr),F_Osc);
    Serial.print(String(F_Osc).substring(0,2) + "." + String(F_Osc).substring(2,11)); Serial.println("
MHz");
  }else
  {
    if (inputString.startsWith("STOP")){
      FreqMeasure.end();
      FreqMeasureActive = false;
      Serial.println("OK");
    }else
    {
      if (inputString.startsWith("STAR")){
        FreqMeasure.begin();
        FreqMeasureActive = true;
        first = true;
        Np = 0;
        Ncp = 0;
        Serial.println("OK");
        Serial << "Medicion de Frecuencia y Periodo para " << F_Osc << " Mhz" << endl;
        Serial.println("Np,Ncp,Periodo,Frecuencia");
      }else{
        if (inputString.startsWith("NPAV") && inputString.length()<=5){
          Serial.print("Np_average: "); Serial.println(Np_Average);
        }else
        {
          if (inputString.startsWith("NPAV") && inputString.length()==10){
            Np_AverageString = inputString.substring(4,11);
            Np_Average = Np_AverageString.toInt();
            //Escribe el valor de Np_Average para la EEPROM
            EEPROM.writeInt(pgm_read_byte(&Np_Average_Addr),Np_Average);
            Serial.print("Np_average: "); Serial.println(Np_Average);
          }else
          {
            Serial.println("ERROR");
          }
        }
      }
    }
  }
  if (!FreqMeasureActive) Serial.print(">");

  // clear the string:
  inputString = "";
  stringComplete = false;
}
if (FreqMeasureActive){
  if (Np < Np_Average){
    if (FreqMeasure.available()) {
      if (first==false){
        Np++;
        Ncp += FreqMeasure.read();
      }else {
        Ncp = FreqMeasure.read();
        Ncp = 0;
        first=false;
      }
    }
  }
}
}
}
}

```

```

//Calculo de la frecuencia y el periodo
if (Np==Np_Average){
  Frequency = (float)Np / Ncp * F_Osc;
  Period = (float) Ncp / (Np * F_Osc); //1 / Frequency;

  //transmite los resultados
  Serial << Np << "," << Ncp << "," ;
  Serial.print(Period, 9);Serial.print(",");Serial.println(Frequency,9);
  //Reset counter;
  Ncp = 0;
  Np =0;
}
}

}

/*
SerialEvent ocurre cada vez que un nuevo dato arriba por la entrada RX.
Esta funcion corre entre cada una de las ejecuciones del loop()
*/
void serialEvent() {
  while (Serial.available()) {
    // Obtener el nuevo byte
    char inChar = (char)Serial.read();
    // adicionarlo al final de la cadena de entrada
    inputString += inChar;
    // si el caracter recibido es return, activa la bandera
    // de manera que en el loop se ejecute algo en consecuencia
    if (inChar == '\r') {
      stringComplete = true;
    }
  }
}
}

```

Anexo 4

Generador de funciones Leader LFG-1300S (extracto).

1. DESCRIPTION

The LFG-1300S Figure 1, is an extremely versatile signal source for design, development and service applications.

It provides a choice of sine, triangle, sawtooth, square and pulse signals over the frequency range of 0.002 Hz to 2 MHz. Voltage control of the frequency of the master oscillator permits linear or logarithmic sweep using internal

or externally-applied sweep control. In addition, an internal modulator permits amplitude modulation with the option of a suppressed-carrier mode. Calibrated control of the output level is provided, as well as variable dc offset. A fixed TTL output is also provided. \

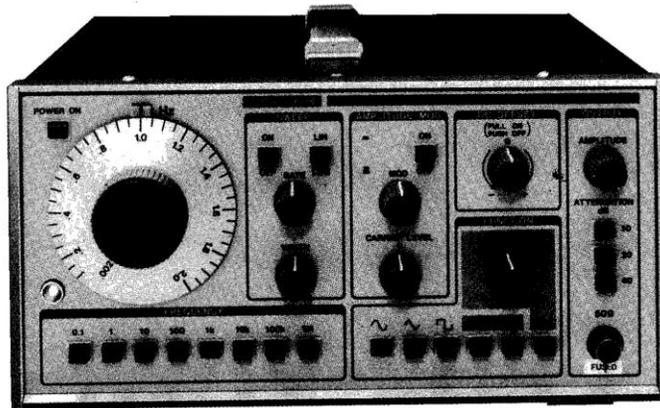


Figure 1. LFG-1300S Function Generator

2. FEATURES

1. Wide frequency range, 0.002 Hz to 2 MHz in eight ranges.
2. Choice of output waveform includes sine, triangle, sawtooth square and pulse signals. In addition, control of dc offset permits the superimposition of dc levels on the output signal.
3. A separate TTL output with a fan-out of 20 permits direct drive of TTL logic circuits.
4. Low distortion, a THD of 0.5% or less is maintained throughout the frequency range.
5. Built in swept frequency function provides a choice of linear or logarithmic sweep with variable sweep width and rate. A sawtooth for horizontal axis drive to an oscilloscope is provided.
6. External sweep-control voltage applied at the VCG connector, permits external control of frequency, sweep, or frequency modulation of the master oscillator.
7. An internal amplitude modulator permits both AM and double-sideband suppressed-carrier modulation for communications and instrumentation applications.
8. Step attenuators provide 10, 20, and 40 dB steps for a total attenuation of 70 dB into a 50 ohm load. An output control provides continuous control of signal level into the attenuator.

3. SPECIFICATIONS

Frequency Range	(0.02 Hz–2 MHz in 8 ranges, uncalibrated to 0.002 Hz): 0.02 Hz–0.2 Hz. 0.2 Hz–2 Hz. 2 Hz–20 Hz. 20 Hz–200 Hz. 200 Hz–2 kHz. 2 kHz–20 kHz. 20 kHz–200 kHz. 200 kHz–2 MHz.	DC Level	Controlled by dc Offset: ± 10 V.
Accuracy	0.02 Hz to 200 kHz: $\pm 3\%$ rdg, $\pm 3\%$ f.s. 200 kHz to 2 MHz: $\pm 5\%$ rdg, $\pm 5\%$ f.s.; for sawtooth	Sweep Capabilities	
Waveforms		Type	Linear or Logarithmic.
Sine wave		Rate (duration)	0.2 Hz to 50 Hz (5 s to 20 ms).
Voltage	20 V p-p (7 V rms) open circuit.	Width	1,000: 1 max, continuously variable.
Distortion	10 Hz–20 kHz; $< 0.5\%$ 20 kHz–100 kHz; $< 1\%$ 100 kHz–2 MHz; $< 3\%$	Ramp Output (for oscilloscope H-input)	0 to +10 V.
Flatness	0.02 Hz–2 MHz within ± 0.3 dB.	AM Capabilities	
Triangle		Modulation Level	0 to 100%.
Voltage	20 V p-p open circuit.	Carrier Level	Adjusted by front panel control.
Symmetry	1% (0.02 Hz to 100 kHz).	Output Level Control	
Sawtooth		Attenuator	10, 20, 40 dB (0-70 dB, 10 dB steps).
Voltage	20 V p-p open circuit.	Impedance	50 Ω .
Symmetry	15:85 or 85:15 fixed.	Max Level	20 V p-p adjustable.
Square Wave Output		Rear Panel Inputs/Outputs VCO	Input for external frequency control signal.
Voltage:	20 V p-p open circuit.	Mod	Input for AM signal.
Symmetry	1% (0.02 Hz to 100 kHz).	GCV	Output for oscilloscope H-Axis.
Rise Time	Less than 100 ns.	TTL	Fixed level TTL output, fan out = 20.
Pulse		Physical	
Voltage	20 V p-p open circuit.	Size (WxHxD)	250 X 125 X 250 mm.
Symmetry	9:1–1:9 Continuously Variable.	Weight	9 lbs, 4 kg approx.
TTL Output Fan Out	20 TTL.	Power Requirements	100, 117, 200 or 234 V ac, 50–60 Hz.

Anexo 5

Especificaciones de los contadores de la serie Nova.

The Nova series of counter-timers

SPECIFICATION

	Nova 200	Nova 200X	Nova 2400	Nova 2400X
Frequency range	10Hz – 200MHz (2 ranges)		10Hz – 2.4GHz (3 ranges)	
Timebase	200m sec.			
Time between measurements	10MHz			
Crystal frequency	10MHz			
Stability	< 0.5ppm	< 0.2ppm	< 0.5ppm	< 0.2ppm
Temperature stability	±10ppm – 10°C to +70°C	±1ppm – 0°C to 50°C	±10ppm – 10°C to +70°C	±1ppm – 0°C to 50°C
Aging	<±5ppm/year	<±1ppm/year	<±5ppm/year	<±1ppm/year

Frequency (Input A)

Range	10Hz – 20MHz
Input impedance	1MΩ
Sensitivity	10mVrms sinewave 20Hz–20MHz; 20mVrms sinewave 10Hz–20Hz
Max. input voltage	28V DC, 20V rms
Resolution	100Hz @ 0.01s gate time 1Hz @ 1s gate time 10Hz @ 0.1s gate time 0.1Hz @ 10s gate time
Attenuation	Automatic gain control from 10mV to 400mVrms, then diode clipping
Measurement Accuracy	± (1 count + timebase accuracy)

Frequency (Input B)

Range	15MHz – 200MHz
Input impedance	50Ω
Sensitivity	25mV rms sinewave 15MHz – 200MHz
Max. input voltage	28V DC, 20V rms
Resolution	1KHz @ 0.01s gate time 10Hz @ 1s gate time 100Hz @ 0.1s gate time 1Hz @ 10s gate time

Frequency (Input C)

Range	150MHz – 2.4GHz
Input impedance	50Ω
Sensitivity	10mV rms sinewave 150MHz – 1.3GHz 20mV rms sinewave 1.3GHz – 2GHz 50mV rms sinewave 2GHz – 2.4GHz
Max. input voltage	28V DC, 20V rms @ 50Hz RF INPUT POWER + 7dBm MAX
Resolution	10KHz @ 0.0128s 100Hz @ 1.28s 1KHz @ 0.128s 10Hz @ 12.8s
Measurement accuracy	±(1 count + timebase accuracy)

Period (Input A)

Range	10Hz – 5MHz
Min pulse width	60ns
Measurement type	Single cycle and multiple period average
Display	µs
Period averaged over	1, 10, 100, 1000 cycles
Resolution	100ns + no. of cycles averaged
Measurement accuracy	±(1 count + timebase accuracy + (trigger error ÷ no of cycles averaged))

	Nova 200	Nova 200X	Nova 2400	Nova 2400X
Trigger error	Trigger error = peak to peak noise voltage / Signal slope (V/s)			
	Trigger error for sinewave = $\frac{1}{\text{FREQ} \times \pi \times \text{S/N RATIO}}$			

Count (Input A)

Range	DC – 5MHz TTL Levels
Count max.	(2 x 10 ⁹) – 1
Resolution	1 count
Reset	Manual (reset button)

External timebase

External oscillator in/internal oscillator out; switch selectable	
Calibration frequency	10MHz
Output frequency	10MHz
Input frequency range	DC to 10MHz
Input voltage	0V to + 10V max., HCMOS load
Output drive	45 TTL gates or sink 18mA Source 18mA

Ancillary controls

Display hold; Reset	
---------------------	--

Display

8½ digit 7 segment LCD; Automatic decimal point; Unit indicators for MHz, KHz, us, low Battery, gate, overflow	
--	--

Power requirements

Mains operation	115/230VAC ±10% 50/60Hz, 6VA selectable on panel
-----------------	--

General

Environmental operating range	0°C to 40°C (10% – 80% RH non-condensing)
Case	Robust, lightweight steel painted light grey, with tilt stand
Size	212mm x 228mm x 100mm (Product only) 318mm x 356mm x 141mm (Packed)
Weight	2.6kg (Product only) 2.9kg (Packed)
Supplied accessories	Mains lead, instruction manual, spare fuse
Optional accessories	Passive probes, carry case, service Manual, RF antenna
EMC:	Complies with EN50081-1 and EM50082-1

Black Star reserve the right to alter specifications without notice

Designed and manufactured by

Black★Star

BLACK STAR LIMITED
4 Harding Way, St. Ives, Huntingdon, Cambridgeshire
PE17 4WR England
Telephone: 01480 462440 Fax: 01480 495172



Designed and manufactured in Britain

Available from:



Anexo 6

Resultados de las mediciones de frecuencia.

Tabla 1. Resultados de las mediciones de frecuencia con la placa Arduino Ethernet R3.
Frecuencia patrón: 800 Hz. (fragmento)

Número de pulsos del Oscilador (Ncp)	Número de periodos de la señal (Np)	Frecuencia (F) Hz	Error (F-fp)	Error ² (F-fp) ²
2562181	128	799.920043	-0.079957	0.006393122
2562160	128	799.960021	-0.039979	0.00159832
2562160	128	799.960021	-0.039979	0.00159832
2562158	128	799.960021	-0.039979	0.00159832
2562153	128	799.960021	-0.039979	0.00159832
2562161	128	799.960021	-0.039979	0.00159832
2562171	128	799.960021	-0.039979	0.00159832
2562166	128	799.960021	-0.039979	0.00159832
2562157	128	799.960021	-0.039979	0.00159832
2562164	128	799.960021	-0.039979	0.00159832
2562160	128	799.960021	-0.039979	0.00159832
2562162	128	799.960021	-0.039979	0.00159832
2562174	128	799.960021	-0.039979	0.00159832
2562180	128	799.920043	-0.079957	0.006393122
2562166	128	799.960021	-0.039979	0.00159832
·	·	·	·	·
·	·	·	·	·
·	·	·	·	·
2562158	128	799.960021	-0.039979	0.00159832
2562159	128	799.960021	-0.039979	0.00159832
2562168	128	799.960021	-0.039979	0.00159832
2562163	128	799.960021	-0.039979	0.00159832
2562167	128	799.960021	-0.039979	0.00159832
2562168	128	799.960021	-0.039979	0.00159832
2562166	128	799.960021	-0.039979	0.00159832
2562160	128	799.960021	-0.039979	0.00159832
2562179	128	799.920043	-0.079957	0.006393122
2562166	128	799.960021	-0.039979	0.00159832

n = 3530

$$\sum_{i=1}^n (f_i - f_p)^2 = 14.78895392$$

Desviación Típica Experimental $s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - f_p)^2} = 0.064735558$

Incertidumbre $u = \pm \frac{s}{\sqrt{n}} = 0.001089571$

Tabla 2. Resultado de las mediciones de frecuencia con la placa Arduino Ethernet R3. Frecuencia patrón: 1000 Hz. (fragmento)

Número de pulsos del Oscilador (Ncp)	Número de periodos de la señal (Np)	Frecuencia (F) Hz	Error (F-fp)	Error ² (F-fp) ²
2049651	128	1000.000000	0	0
2049649	128	1000.000000	0	0
2049649	128	1000.000000	0	0
2049655	128	1000.000000	0	0
2049659	128	1000.000000	0	0
2049650	128	1000.000000	0	0
2049661	128	1000.000000	0	0
2049655	128	1000.000000	0	0
2049659	128	1000.000000	0	0
2049657	128	1000.000000	0	0
2049662	128	1000.000000	0	0
2049656	128	1000.000000	0	0
2049680	128	999.937561	-0.062439	0.00389863
2049681	128	999.937561	-0.062439	0.00389863
2049698	128	999.937561	-0.062439	0.00389863
2049676	128	999.937561	-0.062439	0.00389863
2049661	128	1000.000000	0	0
2049666	128	999.937561	-0.062439	0.00389863
2049664	128	999.937561	-0.062439	0.00389863
2049653	128	1000.000000	0	0
2049659	128	1000.000000	0	0
2049642	128	1000.000000	0	0
2049641	128	1000.000000	0	0
.
.
.
2049661	128	1000.000000	0	0
2049658	128	1000.000000	0	0
2049660	128	1000.000000	0	0
2049663	128	1000.000000	0	0
2049670	128	999.937561	-0.062439	0.00389863
2049670	128	999.937561	-0.062439	0.00389863
2049666	128	999.937561	-0.062439	0.00389863
2049659	128	1000.000000	0	0

n = 4416

$$\sum_{i=1}^n (f_i - f_p)^2 = 10.19101548$$

Desviación Típica Experimental

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - f_p)^2} = 0.048044467$$

Incertidumbre

$$u = \pm \frac{s}{\sqrt{n}} = 0.000722984$$

Tabla 3. Resultado de las mediciones de frecuencia con la placa Arduino Ethernet R3. Frecuencia patrón: 1200 Hz. (fragmento)

Número de pulsos del Oscilador (Ncp)	Número de periodos de la señal (Np)	Frecuencia (F) Hz	Error (F-fp)	Error ² (F-fp) ²
1707459	128	1200.38989	0.089892	0.00808057
1707462	128	1200.38989	0.089892	0.00808057
1707453	128	1200.38989	0.089892	0.00808057
1707462	128	1200.38989	0.089892	0.00808057
1707465	128	1200.38989	0.089892	0.00808057
1707456	128	1200.38989	0.089892	0.00808057
1707440	128	1200.38989	0.089892	0.00808057
1707435	128	1200.38989	0.089892	0.00808057
1707464	128	1200.38989	0.089892	0.00808057
1707452	128	1200.38989	0.089892	0.00808057
1707457	128	1200.38989	0.089892	0.00808057
1707449	128	1200.38989	0.089892	0.00808057
1707453	128	1200.38989	0.089892	0.00808057
1707453	128	1200.38989	0.089892	0.00808057
1707449	128	1200.38989	0.089892	0.00808057
1707459	128	1200.38989	0.089892	0.00808057
1707468	128	1200.38989	0.089892	0.00808057
1707442	128	1200.38989	0.089892	0.00808057
1707446	128	1200.38989	0.089892	0.00808057
1707443	128	1200.38989	0.089892	0.00808057
1707446	128	1200.38989	0.089892	0.00808057
.
.
.
1707446	128	1200.38989	0.089892	0.00808057
1707451	128	1200.38989	0.089892	0.00808057
1707461	128	1200.38989	0.089892	0.00808057
1707450	128	1200.38989	0.089892	0.00808057
1707445	128	1200.38989	0.089892	0.00808057
1707454	128	1200.38989	0.089892	0.00808057
1707442	128	1200.38989	0.089892	0.00808057
1707453	128	1200.38989	0.089892	0.00808057
1707448	128	1200.38989	0.089892	0.00808057
1707443	128	1200.38989	0.089892	0.00808057

n = 2889

$$\sum_{i=1}^n (f_i - f_p)^2 = 23.34477154$$

Desviación Típica Experimental

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - f_p)^2} = 0.089907562$$

Incertidumbre

$$u = \pm \frac{s}{\sqrt{n}} = 0.001672717$$

Tabla 4. Resultado de las mediciones de frecuencia con la placa Arduino Ethernet R3. Frecuencia patrón: 1400 Hz. (fragmento)

Número de pulsos del Oscilador (Ncp)	Número de periodos de la señal (Np)	Frecuencia (F) Hz	Error (F-fp)	Error ² (F-fp) ²
1463870	128	1400.13989	0.139892	0.01956977
1463883	128	1400.13989	0.139892	0.01956977
1463880	128	1400.13989	0.139892	0.01956977
1463884	128	1400.13989	0.139892	0.01956977
1463877	128	1400.13989	0.139892	0.01956977
1463880	128	1400.13989	0.139892	0.01956977
1463875	128	1400.13989	0.139892	0.01956977
1463885	128	1400.13989	0.139892	0.01956977
1463879	128	1400.13989	0.139892	0.01956977
1463891	128	1400.13989	0.139892	0.01956977
1463876	128	1400.13989	0.139892	0.01956977
1463878	128	1400.13989	0.139892	0.01956977
1463881	128	1400.13989	0.139892	0.01956977
1463884	128	1400.13989	0.139892	0.01956977
1463883	128	1400.13989	0.139892	0.01956977
1463869	128	1400.13989	0.139892	0.01956977
1463872	128	1400.13989	0.139892	0.01956977
1463887	128	1400.13989	0.139892	0.01956977
1463874	128	1400.13989	0.139892	0.01956977
.
.
.
1463880	128	1400.13989	0.139892	0.01956977
1463898	128	1400.13989	0.139892	0.01956977
1463874	128	1400.13989	0.139892	0.01956977
1463876	128	1400.13989	0.139892	0.01956977
1463882	128	1400.13989	0.139892	0.01956977
1463874	128	1400.13989	0.139892	0.01956977
1463867	128	1400.13989	0.139892	0.01956977
1463874	128	1400.13989	0.139892	0.01956977
1463881	128	1400.13989	0.139892	0.01956977
1463890	128	1400.13989	0.139892	0.01956977
1463880	128	1400.13989	0.139892	0.01956977
1463864	128	1400.13989	0.139892	0.01956977

n = 2412

$$\sum_{i=1}^n (f_i - f_p)^2 = 43.17589177$$

Desviación Típica Experimental

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - f_p)^2} = 0.133820317$$

Incertidumbre

$$u = \pm \frac{s}{\sqrt{n}} = 0.002724792$$

Tabla 5. Resultado de las mediciones de frecuencia con la placa Arduino Ethernet R3. Frecuencia patrón: 1600 Hz. (fragmento)

Número de pulsos del Oscilador (Ncp)	Número de periodos de la señal (Np)	Frecuencia (F) Hz	Error (F-fp)	Error ² (F-fp) ²
1280923	128	1600.07996	0.079956	0.00639296
1280925	128	1600.07996	0.079956	0.00639296
1280913	128	1600.07996	0.079956	0.00639296
1280915	128	1600.07996	0.079956	0.00639296
1280919	128	1600.07996	0.079956	0.00639296
1280921	128	1600.07996	0.079956	0.00639296
1280917	128	1600.07996	0.079956	0.00639296
1280918	128	1600.07996	0.079956	0.00639296
1280919	128	1600.07996	0.079956	0.00639296
1280911	128	1600.07996	0.079956	0.00639296
1280910	128	1600.07996	0.079956	0.00639296
1280920	128	1600.07996	0.079956	0.00639296
1280921	128	1600.07996	0.079956	0.00639296
1280916	128	1600.07996	0.079956	0.00639296
1280927	128	1600.07996	0.079956	0.00639296
1280921	128	1600.07996	0.079956	0.00639296
1280917	128	1600.07996	0.079956	0.00639296
1280920	128	1600.07996	0.079956	0.00639296
1280920	128	1600.07996	0.079956	0.00639296
1280916	128	1600.07996	0.079956	0.00639296
.
.
.
1280920	128	1600.07996	0.079956	0.00639296
1280916	128	1600.07996	0.079956	0.00639296
1280916	128	1600.07996	0.079956	0.00639296
1280914	128	1600.07996	0.079956	0.00639296
1280923	128	1600.07996	0.079956	0.00639296
1280913	128	1600.07996	0.079956	0.00639296
1280911	128	1600.07996	0.079956	0.00639296
1280907	128	1600.07996	0.079956	0.00639296
1280912	128	1600.07996	0.079956	0.00639296
1280915	128	1600.07996	0.079956	0.00639296
1280912	128	1600.07996	0.079956	0.00639296

n = 1455

$$\sum_{i=1}^n (f_i - f_p)^2 =$$

9.301759617

Desviación Típica Experimental

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - f_p)^2} = 0.07998349$$

Incertidumbre

$$u = \pm \frac{s}{\sqrt{n}} = 0.002096857$$

Tabla 6. Resultado de las mediciones de frecuencia con la placa Arduino Ethernet R3. Frecuencia patrón: 1800 Hz. (fragmento)

Número de pulsos del Oscilador (Ncp)	Número de periodos de la señal (Np)	Frecuencia (F) Hz	Error (F-fp)	Error ² (F-fp) ²
1138591	128	1800.11243	0.112426	0.01263961
1138588	128	1800.11243	0.112426	0.01263961
1138583	128	1800.11243	0.112426	0.01263961
1138586	128	1800.11243	0.112426	0.01263961
1138600	128	1800.11243	0.112426	0.01263961
1138600	128	1800.11243	0.112426	0.01263961
1138612	128	1800.11243	0.112426	0.01263961
1138598	128	1800.11243	0.112426	0.01263961
1138587	128	1800.11243	0.112426	0.01263961
1138593	128	1800.11243	0.112426	0.01263961
1138590	128	1800.11243	0.112426	0.01263961
1138597	128	1800.11243	0.112426	0.01263961
1138599	128	1800.11243	0.112426	0.01263961
1138589	128	1800.11243	0.112426	0.01263961
1138576	128	1800.11243	0.112426	0.01263961
1138595	128	1800.11243	0.112426	0.01263961
1138579	128	1800.11243	0.112426	0.01263961
1138577	128	1800.11243	0.112426	0.01263961
1138592	128	1800.11243	0.112426	0.01263961
.
.
.
1138570	128	1800.11243	0.112426	0.01263961
1138588	128	1800.11243	0.112426	0.01263961
1138586	128	1800.11243	0.112426	0.01263961
1138586	128	1800.11243	0.112426	0.01263961
1138591	128	1800.11243	0.112426	0.01263961
1138589	128	1800.11243	0.112426	0.01263961
1138594	128	1800.11243	0.112426	0.01263961
1138585	128	1800.11243	0.112426	0.01263961
1138587	128	1800.11243	0.112426	0.01263961
1138586	128	1800.11243	0.112426	0.01263961
1138590	128	1800.11243	0.112426	0.01263961
1138591	128	1800.11243	0.112426	0.01263961

n = 2855

$$\sum_{i=1}^n (f_i - f_p)^2 =$$

36.25901643

Desviación Típica Experimental

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - f_p)^2} = 0.112714821$$

Incertidumbre

$$u = \pm \frac{s}{\sqrt{n}} = 0.002109492$$

Tabla 7. Resultado de las mediciones de frecuencia con la placa Arduino Ethernet R3. Frecuencia patrón: 2000 Hz. (fragmento)

Número de pulsos del Oscilador (Ncp)	Número de periodos de la señal (Np)	Frecuencia (F) Hz	Error (F-fp)	Error ² (F-fp) ²
1024754	128	2000.24988	0.249877	0.06243852
1024751	128	2000.24988	0.249877	0.06243852
1024757	128	2000.24988	0.249877	0.06243852
1024773	128	2000.00000	0	0
1024765	128	2000.24988	0.249877	0.06243852
1024760	128	2000.24988	0.249877	0.06243852
1024752	128	2000.24988	0.249877	0.06243852
1024767	128	2000.24988	0.249877	0.06243852
1024773	128	2000.00000	0	0
1024759	128	2000.24988	0.249877	0.06243852
1024760	128	2000.24988	0.249877	0.06243852
1024764	128	2000.24988	0.249877	0.06243852
1024763	128	2000.24988	0.249877	0.06243852
1024761	128	2000.24988	0.249877	0.06243852
1024761	128	2000.24988	0.249877	0.06243852
1024760	128	2000.24988	0.249877	0.06243852
1024768	128	2000.00000	0	0
1024761	128	2000.24988	0.249877	0.06243852
1024766	128	2000.24988	0.249877	0.06243852
1024764	128	2000.24988	0.249877	0.06243852
1024766	128	2000.24988	0.249877	0.06243852
1024758	128	2000.24988	0.249877	0.06243852
.
.
.
1024760	128	2000.24988	0.249877	0.06243852
1024761	128	2000.24988	0.249877	0.06243852
1024756	128	2000.24988	0.249877	0.06243852
1024774	128	2000.00000	0	0
1024760	128	2000.24988	0.249877	0.06243852
1024757	128	2000.24988	0.249877	0.06243852
1024756	128	2000.24988	0.249877	0.06243852
1024750	128	2000.24988	0.249877	0.06243852
1024759	128	2000.24988	0.249877	0.06243852

n = 15561

$$\sum_{i=1}^n (f_i - f_p)^2 = 281.2230721$$

Desviación Típica Experimental

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (f_i - f_p)^2} = 0.134437577$$

Incertidumbre

$$u = \pm \frac{s}{\sqrt{n}} = 0.00107771$$

Anexo 7

Fuentes de incertidumbre total asociada al patrón de referencia.

Tabla 8. Fuentes de incertidumbre asociadas al patrón de referencia.
Frecuencímetro Nova 200.

Método medición: Contador directo				Frecuencias (Hz)							
Base de tiempo			u	800	1000	1200	1400	1600	1800	2000	
Frecuencia	10	1.00E+07	Hz	Estabilidad de la base de tiempo							
Estabilidad	0.5	1.00E-06	ppm								
temp	10	1.00E-06	ppm 10-70°C								
envejecimiento	5	1.00E-06	Ppm/año								
Resolución	0.1		Hz @ 10seg								
				4.0E-04	5.0E-04	6.0E-04	7.0E-04	8.0E-04	9.0E-04	1.0E-03	
				8.0E-03	1.0E-02	1.2E-02	1.4E-02	1.6E-02	1.8E-02	2.0E-02	
				4.0E-03	5.0E-03	6.0E-03	7.0E-03	8.0E-03	9.0E-03	1.0E-02	
				0.029	0.029	0.029	0.029	0.029	0.029	0.029	
Error ± 1 cuenta				± 1 cuenta	0.00125	0.001	0.00083	0.00071	0.00062	0.00056	0.0005
Incertidumbre Total = ± (1 cuenta + estabilidad base tiempo)					0.03025	0.03098	0.03185	0.03285	0.03398	0.03521	0.03653

Anexo 8

Software para la unidad de adquisición de datos basado en Arduino Ethernet R3

```
/*
*****
*   Unidad de Adquisición Arduino Ethernet |R3
*****
#include <FreqMeasure.h>
#include <Streaming.h>
#include <EEPROMex.h>
#include "Arduino.h"
#include <avr/pgmspace.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include "RTClib.h"
#include "VarEEPROM.h" //Direcciones EEPROM de los valores de la configuracion

//constantes para el numero de filas y columnas en la LCD
const int numRows = 2;
const int numCols = 16;
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 6, 5, 4, 3);

//Variables utilizadas para los comandos de consola por el puerto serie
String inputString = ""; // Cadena para los datos de entrada
boolean stringComplete = false; // verdadero cuando la cadena de datos de entrada ha finalizado
con un CR
String F_OscString = ""; // cadena valor ajustado de la frecuencia del reloj (oscilador)
String Np_AverageString = ""; // cadena número de periodos a promediar en la medicion de
frecuencia

//Variables para la medición de frecuencia
boolean FreqMeasureActive = false; // Activar el proceso de medición de frecuencia
unsigned long Ncp = 0; // Numero de pulsos de reloj contados (Number clock's pulses)
unsigned int Np = 0; // Numero de periodos de la señal de entrada
float Period, Frequency; // Período y frecuencia calculados

unsigned long F_Osc = 16000000; // Frecuencia de reloj
unsigned int Np_Average = 128; // Numero de periodos a promediar

const int inTimeClockPin = 2; // Pin 2, se monitorean la señal del RTC con T = 1 seg para
generar una interrupcion que dispara el proceso de medicion según el intervalo definido.
const int LinePowerPin = 9; // Pin 9, salida del LED, se utiliza para controlar la alimentación de
la línea de transductores

RTC_DS1307 rtc;

bool SensorStandby = true; // verdadero para indicar primera medición, se descarta el
primer (o 10)intervalo para evitar problemas de ruido, desfasajes, de la señal de entrada
byte SensorQty, SensorLast; // Cantidad de transductores instalados, dirección ultimo
sensor instalado
byte SensorNumber;
```

```

long MeasureInterval;           // Intervalo de medicion en segundos
unsigned long SensorStartTime;  // Momento de inicio de transmision del sensor
DateTime NextMeasureTime;      // Hora de la sigte medición
DateTime StartMeasureTime;     // Hora de inicio de la medición
DateTime FinalMeasureTime;
DateTime now1;

byte count =0;
void setup() {
  Serial.begin(57600);
  lcd.begin(numCols, numRows);

  pinMode(LinePowerPin, OUTPUT);           // Pin 9 - Salida
  digitalWrite(LinePowerPin, HIGH);       // Alimentación de Linea de transductores OFF

  pinMode(inTimeClockPin, INPUT);         // Pin 2 - Entrada (pulsos del RTC cada un
segundo
  attachInterrupt(0, CheckMeasureInterval, RISING); // Manipulador de la interrupcion 0 que se
disparara con el flanco de subida de cada pulso del RTC
  //interrupts();

  Wire.begin();
  rtc.begin();
  Ds1307SqwPinMode mode = SquareWave1HZ;
  rtc.writeSqwPinMode(mode);
  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
  Serial << "DASduino Ver. 1.0" << endl;
  lcd.setCursor(4,0);
  lcd.print("DASduino");
  lcd.setCursor(3, 1);
  lcd.print("Ver 1.0");
  delay(100);
  //Mostrar fecha y hora actual
  lcd.setCursor(0,1);
  DateTime now = rtc.now();
  if (now.day()<10) lcd << "0";
  lcd << now.day() << "/";
  if (now.month()) lcd << "0";
  lcd << now.month() << "/" << now.year() << " ";
  if (now.hour() < 10 ){
    lcd << "0" ;
  }
  lcd << now.hour() << ":";
  if (now.minute() < 10 ){
    lcd << "0";
  }
  lcd << now.minute();

  delay(100);

  lcd.clear();

```

```

lcd << "Standby...";
// Inicializar datos de la configuración
SensorQtty = 8;//EEPROM.read(pgm_read_byte(&SensorQtty_Addr));
SensorLast = 10;EEPROM.read(pgm_read_byte(&SensorLast_Addr));
Np_Average = EEPROM.readInt(pgm_read_byte(&Np_Average_Addr));
F_Osc = EEPROM.readLong(pgm_read_byte(&F_Osc_Addr));
MeasureInterval = 25; //EEPROM.readLong(pgm_read_byte(&HourInterval_Addr)) * 3600L +
EEPROM.readLong(pgm_read_byte(&MinInterval_Addr))* 60;

NextMeasureTime = now.unixtime() + MeasureInterval;
FreqMeasureActive = false;

Serial << "-----" << endl;
Serial << "Start time: ";
if (now.hour() < 10){
  Serial << "0";
};
Serial << now.hour() << "." ;
if (now.minute()<10)Serial << "0";
Serial << now.minute() << ".";
if (now.second()<10){Serial <<"0";};
Serial << now.second() <<endl ;
Serial << endl;

Serial << "SensorQtty: " << SensorQtty << endl;
Serial << "SensorLast: " << SensorLast << endl;
Serial << "MeasureInterval: " << MeasureInterval << endl;
Serial << "NextMeasureTime: ";
if (NextMeasureTime.hour()<10) Serial << "0";
Serial << NextMeasureTime.hour() << "." ;
if (NextMeasureTime.minute()<10) {Serial << "0";};
Serial << NextMeasureTime.minute() << ".";
if (NextMeasureTime.second()<10){Serial << "0";};
Serial << NextMeasureTime.second() << endl ;
Serial << endl;

Serial << "Osc. Freq: " << F_Osc << " MHz" << endl;
Serial << "Qtty Period Average: " << Np_Average << endl;
Serial << endl;

inputString.reserve(25);
now1 = rtc.now();
digitalWrite(LinePowerPin, LOW);
}

void loop() {
if (FreqMeasureActive){
  if (Np < Np_Average || !SensorStandby) {
    if (FreqMeasure.available()) {
      Np++;
      if (SensorStandby){
        Ncp = FreqMeasure.read();
        Ncp = 0;
        // Aqui debe ir el codigo para medir la corriente

```

```

// En estos 10 ciclos se debe medir la corriente de trabajo del sensor activo
// se suman los valores medidos se promedian cuando Np = 10
// SensorWorkCurrent += analogRead(SensorCurrentPin);
if (Np == 1) {
    SensorStartTime = millis();
}
if (Np == 10){
    SensorStandby=false;
    Np=0;
    //DateTime now = rtc.now();
    Serial << "Now - StartMeasureTime: " << now1.unixtime()<< " - " <<
StartMeasureTime.unixtime() << " = " << now1.unixtime() - StartMeasureTime.unixtime() << endl;
    SensorNumber = (now1.unixtime() - StartMeasureTime.unixtime()) / 2;
    //Aqui debe ir el codigo para promediar el valor de la corriente de trabajo del sensor
}
} else {
    Ncp += FreqMeasure.read();
}
}
else{
    if (!SensorStandby){
        unsigned long Dt = millis() - SensorStartTime;
        if (Dt >=1500){
            SensorStandby = true;
            Np = 0;
            Ncp = 0;
            SensorStartTime =0;
        }
    } // !SensorStandby
} // FreqMeasure.available
//Serial.print("SensorStandby: "); Serial.println( SensorStandby ? "TRUE" : "FALSE");
}

//Calculate of frequency and period time
if (Np==Np_Average){
    Frequency = (float)Np / Ncp * F_Osc;
    Period = (float) Ncp / (Np * F_Osc); //1 / Frequency;

//Mostrar resultados en LCD
lcd.setCursor(1,0);
lcd.print("S:");
lcd.print(SensorNumber);
lcd.setCursor(6,0);
lcd.print(Frequency,2);
lcd.print("Hz");
lcd.setCursor(5,1);
lcd.print(Period,6);
//Envia los resultados al puerto serie
Serial << "Sensor Number: " << SensorNumber << endl;
Serial.println("Np,Ncp,Period,Frequency");

Serial.print( Np); Serial.print(",");Serial.print(Ncp);Serial.print(",");
Serial.print(Period, 9);Serial.print(",");Serial.println(Frequency,9);
}
}
}

```

```

}

/*****
* Manipulador de la interrupcion 0 (Pin 2)
/* Esta funcion se ejecuta con cada flanco de subida de la señal recibida por el pin 2
* que corresponde a la interrupcion de hardware 0 del microcontrolador
*****/

void CheckMeasureInterval(){
  noInterrupts();
  now1 = now1.unixtime()+1;

  Serial.print("Interrupcion: "); Serial.println(++count);

  Serial.print("RTC Interruption at: "); Serial.println(now1.unixtime());
  if (now1.unixtime() >= NextMeasureTime.unixtime()){
    FreqMeasureActive = true;          // Activar la medicion de frecuencia
    NextMeasureTime = NextMeasureTime + MeasureInterval;    // Actualiza nueva hora de
    medicion

    Serial.println("FreqMeasureActive: ON");

    digitalWrite(LinePowerPin, HIGH);          // Alimentación de Linea de transductores ON
    StartMeasureTime = now1;                    // Hora de inicio de la medición
    Serial.print("StartMeasureTime: "); Serial.println(StartMeasureTime.unixtime());
    Serial.print("NextMeasureTime: "); Serial.println(NextMeasureTime.unixtime());
    FinalMeasureTime = StartMeasureTime + 2 * SensorLast + 2; // Hora Final
    Serial.print("FinalMeasureTime: "); Serial.println(FinalMeasureTime.unixtime());
    FreqMeasure.begin();                       // Inicia proceso de medición
    Np = 0;
    Ncp =0;
  }else {
    if (now1.unixtime() >= FinalMeasureTime.unixtime() && FreqMeasureActive){
      FreqMeasureActive = false;              // Desactiva la Medicion de frecuencia
      FreqMeasure.end();                       // Finaliza proceso de medición
      digitalWrite(LinePowerPin, LOW);        // Alimentación de Linea de transductores OFF
      StartMeasureTime = 0;                   // Resetea Hora de inicio de la medición y fin de la
    medicion
      FinalMeasureTime = 0;
      SensorStandby = true;
      Serial.println("FreqMeasureActive: OFF");
      Serial.print("NextMeasureTime: "); Serial.println(NextMeasureTime.unixtime());
    }
  }
}

interrupts();
}

```

