



Pontificia Universidad Javeriana

Trabajo Final de Maestría

**Diseño e Implementación de un Sistema Inteligente
de Medición y Monitoreo de Energía Eléctrica
Trifásica para el Hogar**

James Fernando Zafra Martínez

Noviembre 16 2017

1	INTRODUCCIÓN.....	4
2	MARCO TEÓRICO	7
2.1	Metodología Establecida Para el Diseño.....	7
2.2	Trabajos y Desarrollos Previos	9
2.3	Propuesta de Implementación	9
3	OBJETIVOS ESPECIFICACIONES Y NORMAS.....	11
3.1	Objetivos.....	11
3.1.1	Objetivo Principal.....	11
3.1.2	Objetivos Específicos.....	11
3.2	Especificaciones Eléctricas de Medición	12
3.3	Especificaciones de Comunicaciones:	13
3.4	Especificaciones de Interface de Usuario.....	13
4	ARQUITECTURA PROPUESTA.....	14
4.1	Etapas del Diseño:.....	14
4.1.1	Acondicionamiento de Señales.....	14
4.1.2	Conversión Análoga/Digital o Toma de Muestras.....	14
4.1.3	Almacenamiento de muestras:	14
4.1.4	Cálculo de parámetros y acumulación de datos:	15
4.1.5	Comunicación e interfaces:.....	15
4.2	Plataforma de desarrollo:.....	15
4.3	Arquitectura Propuesta	16
5	DESARROLLO E IMPLEMENTACIÓN.....	18

5.1	Implementación del hardware. Etapa 1.....	18
5.1.1	Muestreo de Voltaje	18
5.1.2	Muestreo de Corriente	20
5.1.3	Filtro Anti – Aliasing	24
5.1.4	Muestreo y Conversión.	25
5.1.5	Selección del ADC.....	27
5.1.6	Detector de conexión de fases.....	28
5.1.7	Detector de cruce por cero de Voltaje	30
5.1.8	Módulo de transmisión ZigBee.	30
5.2	Implementación de Módulos de Software	33
5.2.1	Proceso de Background	34
5.2.1.2	Sub-Buffering. Método propuesto. Etapa 2	35
5.2.1.1	Módulo de Toma de Muestras y Timers. Etapa 2	37
5.2.1.2	Módulo de Lectura datos V,I. Etapa 3	38
5.2.2	Proceso de Foreground. Etapa 4.....	40
5.2.2.1	Módulo de Cálculos	42
5.2.2.2	Módulo de Control.....	45
5.2.3	Módulo de Interface a Usuario. Etapa 5	45
5.2.4	Módulo de Comunicaciones ZigBee y serial. Etapa 5.....	45
6	ANALISIS DE RESULTADOS	48
6.1	Adecuación de señales: Completar con muestras de osciloscopio	48
6.2	Detección de Fases y sincronismo: Resultado.....	¡Error! Marcador no definido.
6.3	Método sub_buffering: Mostrar diferentes tamaños de buffer implementados:	51
7	CONCLUSIONES.....	56
8	ANEXOS	58
9	BIBLIOGRAFIA Y FUENTES DE INFORMACION	57

1 INTRODUCCIÓN

Atender la demanda de energía, implica no solo hacerla llegar hasta el usuario (Generar, Transmitir y Distribuir), sino medirla y cobrarla. Hasta finales del siglo XX, la mayor parte de países del mundo, realizaban la medición del consumo de energía mediante contadores mecánicos de energía eléctrica conocidos como contadores tipo Ferraris [1, 2, 3]. Ello desde luego implicaba (e implica todavía en muchos países) el desplazamiento de personal para tomar las respectivas lecturas de consumo y la posterior consolidación de datos.

Este sistema de recolección de información, cambió con el advenimiento de los medidores electrónicos, dando lugar al llamado Automated Meter Reading (AMR) System, el cual utiliza un sistema de comunicación de una vía (lectura del dispositivo). Pero fueron surgiendo otras necesidades importantes como la programación remota (por cambios de tarifa por ejemplo) y la obtención de datos de calidad de la energía (Power Quality) los cuales necesitaban de un sistema de comunicaciones de doble vía y por tanto requerían de un medidor inteligente capaz de cumplir esta funcionalidad. De esta forma se dio origen al Advanced Metering Infrastructure (AMI) System, que es un sistema con medidores electrónicos de energía con comunicación de doble vía [1, 2, 3].

Los medidores de energía eléctrica, han evolucionado, pasando de aquellos de tipo electromecánico a los electrónicos, sin partes móviles y en la actualidad a medidores autónomos e inteligentes [1]. Los Smart Meter, que son dispositivos electrónicos de medición de energía con capacidad de comunicación [1,2], son relativamente nuevos en el mercado [1,2]. Las aplicaciones de estos dispositivos o sistemas, van desde la simple obtención de datos de consumo por parte de la empresa prestadora de servicios o del usuario, hasta el control de flujo de carga, la venta de energía pre-pagada, el control de la calidad de la energía, el ahorro de energía, la detección de robo de energía, la automatización de hogares, etc. [2].

Paralelamente, se ha dado un gran auge, de fuentes renovables de energía como la solar y la eólica impulsadas por importantes subsidios a su generación, el cual, ha llevado a la imperiosa necesidad de crear una red de distribución inteligente, capaz de medir ya no solo la energía que los usuarios consumen, sino también la que inyectan a la red. Esto ha impulsado el Smart Grid [1, 3, 4, 5], que es un sistema con flujo de energía e información en dos direcciones. Sistema Inteligente.

Como producto de estos cambios hacia los sistemas inteligentes, nacieron los Smart Meters [2, 4, 5] que son los dispositivos de medición de energía con comunicación de doble vía como se ha dicho.

El uso masivo del Smart Meter, en el hogar, está asociado a dos aspectos básicos [2,5].

- La transmisión de los datos, que depende de la tecnología más conveniente según el país: Radio Frecuencia (RF) o Power Line Carrier (PLC))
- El costo del Smart Meter: En este aspecto, el desarrollo tecnológico, ha permitido el diseño de MCUs (microcontroladores) y SoC (System on Chip), de muy bajo costo, que han permitido la masificación de los Smart Meter. Hoy, países como el Reino Unido, Alemania, Finlandia, Dinamarca, Italia, USA, Japón, China y otros, están trabajando fuertemente en la consolidación de su uso [4,5].

Para las empresas prestadoras del servicio de energía, gas y agua, la reducción de costos asociada a la medición de ese servicio y al uso inteligente de recursos es muy importante y en ese sentido, los Smart Meters juegan un papel importante, porque reducen los costos de recolección de datos de consumo y porque inducen al usuario a hacer uso racional del recurso energético. Hoy estos dispositivos son una realidad a nivel mundial y se están instalando en casi toda Europa, USA, China, Japón y otros países, aprovechando la reducción en su costo [2,4,10].

El dispositivo inteligente, para monitorear la energía eléctrica, dentro del hogar, es el que se conoce como In Home Display (IHD) [4, 5] o de forma más general los Home Energy Management System (HEMS), que ya tienen capacidad de interconexión local y remota [5, 11]. Ambos pueden tener integrado un Smart Meter o recibir información de él. Como bien lo expresa Meera Balakrishnan [13], los grandes impulsores del Home Energy Management son: El crecimiento de la demanda de la energía, el control de su uso, el acceso remoto al hogar y los facilitadores tecnológicos (LAN/WAN, ZigBee, Wi-Fi, Z-Wave, etc.) y por supuesto monitorear y controlar el uso de energía, agua y gas.

Todo esto implica, disponer de dispositivos inteligentes dentro del hogar y desarrollar una red inteligente que los interconecte, es decir una HAN (Home Area Network), que a su vez los comuniquen con el mundo exterior. Se estima que el crecimiento de estas redes pasó de 1.5 millones de hogares en el 2009 a 14.7 millones en el 2014, [5,14].

Para la fecha, ya grandes compañías como Google y Microsoft están incursionando en el tema del Smart Energy a través de plataformas como android@home y Microsoft@Hohm y por otro lado grandes fabricantes de electrónica de consumo como Samsung y LG han anunciado, que sus próximos modelos de lavadoras y secadoras podrán programarse desde el celular.

Los más recientes desarrollos en tecnología aplicadas a Smart Meter, incluyen los SoC (System On Chip), los cuales son circuitos integrados de propósito específico para medición de energía eléctrica. Ejemplos de ellos son el MSP430AFE2x3 y MSP430677x de Texas Instruments [6,7] y el ADE 7878 de Analog Devices [8], los cuales vienen con el hardware y el software necesarios para realizar estas medidas, embebidos en un circuito integrado. Normalmente estos SoC, vienen sin un micro controlador adicional, embebido en la misma pastilla, siendo necesario adicionar uno de forma externa, para manejar las comunicaciones e interfaces, necesarias para manejar los datos arrojados por el SoC

Colombia, representa un mercado aun no desarrollado en el campo de los Smart Meters, para el hogar o de tipo industrial. No se conocen desarrollos de aparato de estas características o similar a un IHD o a un HEMS construidos localmente y nuestro estado actual de conocimiento de esta tecnología, permite su diseño y desarrollo sin restricciones. Si agregamos el hecho de que el Smart Meter, puede hacer parte de una red local, su futura expansión se hace interesante, si pensamos por ejemplo en un edificio inteligente, que desee medir todos los parámetros y consumos por piso y consolidar datos para toda la edificación, o una industria donde se desee tener control del consumo por áreas, o el diseño de tableros eléctricos inteligentes,. Todo estos representan amplios segmentos de aplicación.

El presente proyecto, implementa un medidor inteligente de energía eléctrica trifásica para el hogar, basado en un microcontrolador, con capacidad de operar de forma autónoma y adicionalmente que tiene capacidad de comunicación.

El sistema puede medir parámetros básicos de la red eléctrica como voltaje, corriente, frecuencia, calcular los consumos de potencia y energía y el costo asociado a los mismos. A su vez tiene la capacidad de acumular datos de consumo y transmitirlos localmente (dentro de la casa) de forma inalámbrica hacia un punto de recolección utilizando el estándar 802.15.4, que es el más utilizado para este tipo de aplicaciones, por su bajo consumo[12].

Las mediciones de parámetros y cálculos de consumo, se realizan con un MCU(microcontrolador) mediante tareas de software, embebidas en el MCU y que forman parte del presente diseño.

La comunicación inalámbrica, puede adaptarse en un futuro, para transmitir los datos de consumo a un punto de recolección, desde donde se puedan transmitir hacia un servidor central, para implementar métodos de facturación remota.

Con el presente trabajo, se pretende mostrar paso a paso, el diseño de los principales bloques de hardware y software, necesarios para implementar un Smart Meter, basado en un Microcontrolador (MCU), sin el uso de un SoC. Ello elimina la dependencia de un SoC, específico, que en muchas ocasiones puede tener más funciones de las que realmente se necesiten en determinada aplicación y también puede resultar más costoso, debido a la necesidad de incluir el MCU para realizar las tareas de comunicación e interfaces de usuario.

El Trabajo, desarrolla completamente el software, de muestreo, medición y cálculo de parámetros, de la red mencionada, como también el diseño del hardware necesario, para el acondicionamiento de las señales de entrada. El diseño se implementa en una plataforma de desarrollo, que contiene el MCU seleccionado para tal fin, con sus respectivos periféricos.

Un aspecto novedoso del presente diseño, consiste en el diseño e implementación de un método de muestreo, consistente en subdividir las muestras en buffers pequeños, que pueden ser guardados en memoria RAM. Es decir que en lugar de acumular todos los datos durante un segundo, o varios ciclos en memoria flash, se acumulan en pequeños buffers , con los cuales se realizan los cálculos, que luego se van guardando para completar el buffer de un segundo.. Este método se explicará en la sección 5.2.1

La sección 2, presenta el marco teórico relacionado con este proyecto, en sección 3, se dan las especificaciones del sistema, la sección 4 expone la arquitectura propuesta y la forma de implementarla. En la sección 5 se presentan de forma detallada todos los desarrollos tanto de hardware como de software, necesarios para la implementación. La sección 6, expone los resultados obtenidos.

2 MARCO TEÓRICO

Los Smart Meter, como una parte esencial de los sistemas eléctricos inteligentes (Smart Grid), le permiten a las empresas prestadoras de servicios, obtener información en tiempo real, sobre la demanda de energía eléctrica. Esto facilita la planificación de los sistemas de generación, transmisión y distribución. Las ventajas para los operadores del servicio de energía incluyen[2]: Reducción en costos de recolección de datos, manejo de la demanda, reducción en costos de facturación, etc.

Por otra parte, los usuarios al tener un conocimiento directo y en tiempo real de sus consumos de energía con sus costos asociados, pueden tomar decisiones, para hacer un uso óptimo de la misma, reduciendo sus costos de consumo y contribuyendo al uso racional de la energía.

Para proveer de estas ventajas a usuarios y operadores un Smart meter debe cumplir ciertos requerimientos como precisión, seguridad, e impacto de las señales de radio involucradas. [4].

Por lo anterior, existen normas y estándares, que deben cumplirse para poder construir un Smart Meter, de tal forma que garantice su operación, su funcionalidad y que respete, las normas de seguridad establecidas[4].

Respecto a la precisión, existen dos estándares, reconocidos al respecto: El IEC 62053, IEC62056 y la ANSI 12C.

En cuanto a la seguridad, se debe cumplir con requerimientos de seguridad de la información, ya que esta se realiza de forma remota, siendo importante el cifrado de la misma [4].

Con respecto a la exposición a radiofrecuencia, las normas de emisión en el espectro de frecuencia, están establecidas por la FCC (Federal Communication Commission).

En cuanto a la continuidad en la transmisión de datos el ciclo de trabajo (Duty Cycle) de estos medidores está entre el 1% y el 5%

2.1 Metodología Establecida Para el Diseño

La medición digital de parámetros de una red eléctrica, tales como voltaje, corriente y potencia se realiza mediante la toma periódica de muestras de cada fase durante un intervalo de tiempo. Luego, las muestras tomadas, se almacenan en memoria para realizar los cálculos de valores RMS, Potencia activa, Potencia reactiva, Potencia total, Energía consumida, etc. Los valores calculados se van acumulando y guardando en memoria para posteriormente mostrarlos a través de un display o para transmitirlos a algún otro dispositivo [6,7,8].

El proceso de diseño de un Smart Meter se desarrolla en los siguientes pasos. Ver figura 2.1:

- ✓ Acondicionamiento de las señales de entrada para la toma de muestras: Incluye el diseño de filtros para eliminar ruidos no deseados en la señal y la adecuación de la señal a niveles de voltaje apropiados para su muestreo.

- ✓ Conversión analógica/digital o toma de muestras: Corresponde al proceso de digitalización de la señal, a través de un conversor analógico /digital o ADC y la transferencia de los datos a la CPU para su posterior procesamiento. Normalmente el ADC, está embebido en el MCU.
- ✓ Almacenamiento de muestras: Las muestras digitalizadas, durante un intervalo de tiempo definido, deben guardarse hasta completar el periodo mínimo de toma de muestras (1 segundo normalmente), [15,16,17,18], para proceder a realizar los cálculos respectivos.
- ✓ Cálculo de parámetros y acumulación de datos: Con las muestras obtenidas, de acuerdo a la especificación del sistema, se procede a realizar los cálculos de voltaje rms, corriente rms, potencia, activa, potencia reactiva, potencia total, energía consumida y costos de consumo.
- ✓ Comunicación e interfaces: Es importante poder transmitir los datos obtenidos y calculados fuera del sistema, para su interpretación y toma de decisiones [5,19].

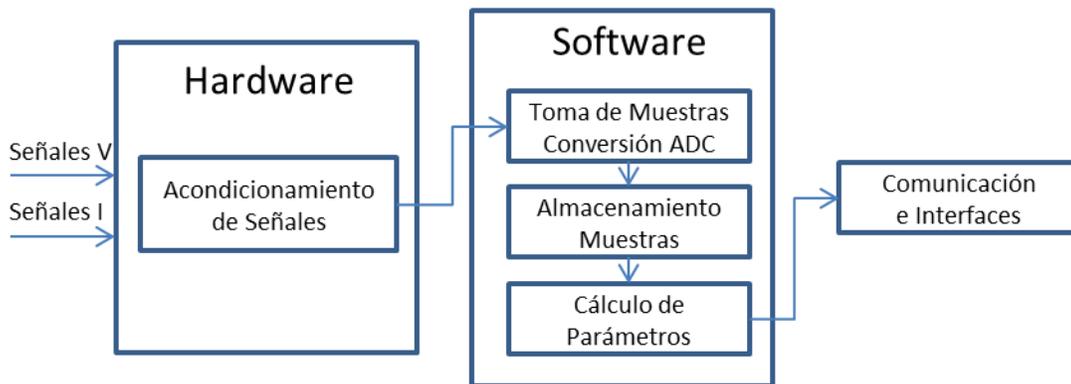


Figura 2.1 Proceso de implementación de un Smart Meter

Para implementar los pasos anteriores un Smart Meter, requiere unas características mínimas tanto de procesamiento, como de inclusión de periféricos.

EL MCU, o procesador que se seleccione, preferiblemente debe poseer unidad matemática con inclusión de algunas funciones básicas de punto flotante (multiplicación, raíz cuadrada y funciones trigonométricas básicas, sin que se requiera un DSP), pero a la vez debe ser de bajo consumo y preferiblemente de bajo costo.

Adicionalmente el MCU debe incluir como mínimo los siguientes periféricos: DMA (Direct Memory Access), que permite la toma continua y periódica de muestras sin intervención de la CPU (o con mínima intervención), varios ADC con varios canales cada uno, Timers de precisión para generar señales de sincronismo, RTC (Real Time Calendar), UART, RS232, Memoria Flash.

Uno de los periféricos indispensables para implementar un Smart Meter, es el ADC. En el caso particular de un Smart Meter trifásico, se necesitarían 7 ADCs independientes para lograr muestrear simultáneamente los tres voltajes, las tres corrientes y la corriente de neutro. Si además, se desea cumplir con algún estándar riguroso como el 62053 o el 62056 clase A - 0.5 [20], que exige

mediciones independientes y con precisión del 0.5%, se necesitaría un MCU con un mínimo de 7 ADCs tipo $\Delta\Sigma$ o incluir los ADCs como periféricos externos al MCU. Esto eleva los costos de un dispositivo de este tipo, implementado de esta forma.

2.2 Trabajos y Desarrollos Previos

Existen varios impulsores del diseño y fabricación de Smart Meters. Por un lado las empresas prestadoras del servicio (utilities) como se ha dicho y también los fabricantes de procesadores, microcontroladores y SoC, quienes promueven su implementación con el desarrollo de medidores prototipo monofásicos y trifásicos, entre ellos Texas Instruments [15,18] , Freesclae (hoy NXP) [16], Microchip Corporation [17]. Todos ellos tendientes a promover sus respectivos dispositivos.

En los desarrollos de estos fabricantes, normalmente se hace un muestreo de las señales, durante 1 segundo y luego se procede a realizar los cálculos.

Varios trabajos independientes, se han realizado en implementación de Smart Meter de energía eléctrica.

Nikhil y Dnyaneshwar [23], implementan un Smart Meter con sistema de tarifación y sistema de detección de robo de energía. (Antitheft), con el uso de un procesador con sistema operativo embebido.

Xuhu, Na, Peihua, Boyang, [24], presentan el diseño de un Smart Meter, con el uso de del SoC AD7778 de Analog Devices, más un MCU de la línea MSP de Texas Instrument, mostrando cómo es necesaria la adición de un MCU, cuando se utiliza un SoC específico de medición de Energía.

Li, Yang, Yu and Zhang [30] Proponen la implementación de un Smart Meter, basado en un DSP. Debe tenerse en cuenta que este tipo de Smart Meter, no cumple el requerimiento de bajo consumo de potencia.

Como puede verse de las referencias mencionadas, existen diferentes tipos de combinaciones de procesadores y periféricos, para desarrollar un Smart Meter. Todas ellas miden los parámetros de la red de forma similar, pero utilizando diferentes procesadores e implementando diferentes métodos de comunicación, de acuerdo a la necesidad.

En cuanto a la metodología, de obtención de muestras y cálculo de parámetros, todas son similares, dividiendo el problema en dos procesos principales, para su implementación: Un proceso de Background o captura de datos y un proceso de Foreground de cálculos, acumulación de datos, comunicaciones e interfaces. La mayoría de los diseños mencionados acumulan los datos durante un segundo para luego realizar los cálculos.

2.3 Propuesta de Implementación

El presente proyecto, implementa un Smart Meter para el hogar o Home Energy Meter (HEM), el cual mide los parámetros fundamentales de una red doméstica, como Voltaje, Corriente, Potencia,

Energía y Costo del consumo, basando su diseño en un microcontrolador de bajo costo, sin el uso de circuitos integrados de medición de energía o SoC.

El sistema implementa, todos los módulos mencionados en la sección 2.1, e ilustrados en la figura 2.1.

Las especificaciones de la red a medir se describen en el capítulo 3

La solución se implementa en una tarjeta de desarrollo, que contiene al MCU seleccionado y los periféricos necesarios. Su descripción detallada se hace en el capítulo 4.

3 OBJETIVOS ESPECIFICACIONES Y NORMAS

3.1 Objetivos

El sistema propuesto, debe cumplir los siguientes objetivos:

3.1.1 Objetivo Principal

- Diseñar y construir un sistema de medición de energía eléctrica trifásica que cumpla con algunas de las especificaciones de los estándares para un Smart meter y que forme parte de una red doméstica de dispositivos inteligentes HAN (Home Area Network).

3.1.2 Objetivos Específicos

- Diseñar e implementar el AFE (Analog Front End), bloque encargado de tomar muestras de voltaje, corriente y frecuencia, el cual es externo a la plataforma.
- Seleccionar el dispositivo de procesamiento a utilizar (DSP o Microcontrolador), con su respectiva plataforma de desarrollo.
- Implementar las tareas de comunicación necesarias, para que el dispositivo forme parte de una HAN (dentro del hogar) y que a su vez se comunique con el exterior (fuera del hogar). Para este proyecto se transmitirá localmente mediante el protocolo 802.15.4 y al exterior por RS485.
- Implementar las tareas necesarias dentro del microcontrolador para calcular los valores de voltaje, corriente, potencia, energía, costos de consumo, sin utilizar circuitos integrados especializados de medición, SoC o ASIC. Esta alternativa de diseño (No Usar SoC), puede incrementar el costo de implementación, pero este costo, es relevante a la hora de producir el dispositivo industrialmente y este no es un objetivo de este trabajo.
- Almacenar datos históricos de consumo y transmitirlos, mediante los protocolos antes mencionados.

El sistema, está diseñado para medir los parámetros fundamentales de una red eléctrica doméstica con consumo de 10KW. Dichos parámetros son: voltaje, corriente, frecuencia, potencia, energía y costo del consumo. El sistema acumula las mediciones, calcula costos de consumo y transmite la información localmente vía inalámbrica utilizando el estándar 802.15.4, base del protocolo ZigBee [4]. También el sistema acumula datos de consumo horario, diario, semanal, mensual y anual y los envía por red inalámbrica utilizando el estándar 802.15.4 en una configuración sencilla de dos nodos, uno de transmisión y uno de recepción.

Las mediciones cumplen con algunas de las especificaciones de los estándares más reconocidos para este propósito, como el IEC 62053-21 para Smart meter clase 2. y la norma técnica Colombiana NTC4569 clase 2, para Smart Meter electrónicos estáticos en cuanto la precisión de las medidas (accuracy) [5]. Como el presente desarrollo, no incluye la fabricación del Smart Meter, muchas de las especificaciones del estándar, no son aplicables. Pero las relacionadas a la precisión de las medidas de voltaje, corriente y potencia, si.

El Smart Meter, implementado corresponde a un medidor estático de energía activa, para el hogar. Esto implica que se pueden tener cargas monofásicas, pero dentro de un sistema trifásico, alimentado de forma balanceada, por esta razón se permiten variaciones del 3% en las mediciones de voltaje y corriente [5].

Las especificaciones se han dividido en tres grupos: Especificaciones eléctricas de medición, especificaciones de comunicaciones y especificaciones de interface de usuario. En las tabla 3.1, 3.2 y 3.3, se detalla cada una de ellas.

3.2 Especificaciones Eléctricas de Medición

Están basadas en el estándar IEC 62053-21 para Smart meter clase 2, en cuanto la precisión de las medidas de voltaje, corriente y potencia.

ESPECIFICACIONE ELÉCTRICAS	
Tipo de red	Trifásica 4 hilos, 3 fases y neutro
Voltaje nominal	120/208 VAC
Máxima variación de voltaje	+/- 15% o $V_{LN} = 102 - 138VAC$
Precisión de la medida de voltaje	3 % IEC62053-21 Clase 2
Corriente nominal por fase	5(30 A). 30 A/fase, 5 ^a neutro en balanceo,
Corriente máxima por fase	30 A
Corriente mínima por fase	300 mA
Precisión de la medida de corriente	3% de la corriente nominal.
Corriente mínima a medir	0.3 A. IEC62053-21 clase 2
Potencia máxima a medir	10 KW 3 fases, 3x 120Vx30A
Medición de Potencia	Por datos acumulados durante 1 segundo
Precisión de la medida de Potencia activa	De acuerdo a precisión en V e I
Medición Potencia reactiva	Por cálculo o por datos acumulados
Precisión de la medida de Potencia reactiva	De acuerdo a precisión en V e I
Frecuencia de red	60 Hz +/- 5%
Precisión de la medida de frecuencia	1.2 Hz

Tabla 3.1 Especificaciones eléctricas del sistema

3.3 Especificaciones de Comunicaciones:

El dispositivo enviará los datos acumulados hacia una terminal RS232. Se dejará diseñada una interface serial tipo R232/RS485 para poder transmitir datos de forma cableada, pero no se implementará la transmisión o recepción de los datos en RS485. Adicionalmente se implementa la comunicación inalámbrica, basada en el estándar 802.15.4, por ser el más utilizado para enviar este tipo de información [4].

ESPECIFICACIONES DE I COMUNICACIONES	
Transmisión Inalámbrica	SI
Protocolo Inalámbrico	802.15.4
Transmisión Cableada	RS232, o, RS 485

Tabla 3.2 Especificaciones de comunicaciones del sistema

3.4 Especificaciones de Interface de Usuario

El sistema, a través de la plataforma de desarrollo, dispone de un display LCD, para visualizar algunos parámetros importantes y de una interface de transmisión serial hacia una terminal, como interfaces de usuario. Los datos que se pueden visualizar, se muestran en la tabla 3.3.

ESPECIFICACIONES DE I NTERFACE DE USUARIO	
Datos a mostrar en el LCD	Parámetros
	Potencia total consumida
Consumo	Consumo de energía última hora en kWh
	Consumo de energía del día en kWh
	Consumo acumulado del mes en kWh
	Consumo acumulado del mes en \$
Parámetros de la red	Voltaje y Corriente RMS en una fase, según capacidad del LCD

Tabla 3.3 Especificaciones de interface de usuario

4 ARQUITECTURA PROPUESTA

El sistema propuesto sigue la metodología expuesta en la sección 2.1 y se realiza en 5 etapas.

4.1 Etapas del Diseño:

Las cinco etapas en que se ha dividido el diseño, se ilustran en la figura 4.1

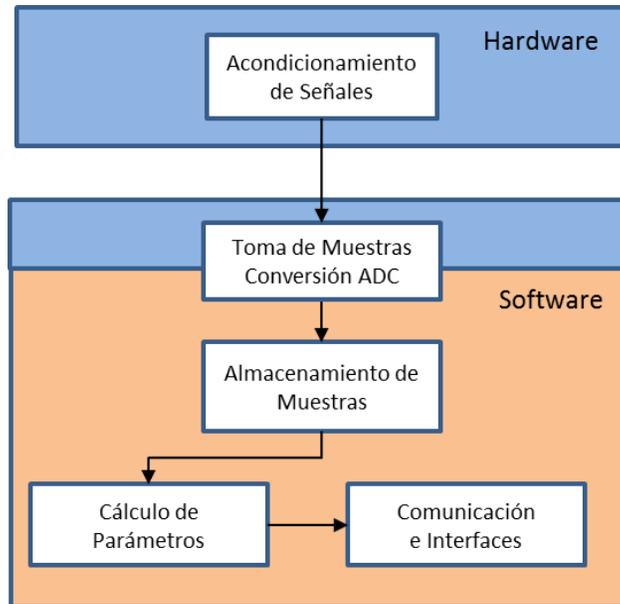


Figura 4.1 Etapas de desarrollo del sistema propuesto

4.1.1 Acondicionamiento de Señales

Es necesario, que las señales de voltaje y corriente de entrada de la red, tengan los niveles adecuados a la entrada del ADC, para poder ser muestreadas. Es decir que los voltajes deben ser reducidos a niveles de 3.3 V. Adicionalmente se necesita incluir un filtro antialiasing, de acuerdo a la frecuencia de muestreo. Finalmente, es importante incluir un circuito de detección de cruce por cero, que es la señal de sincronismo, para el inicio de la toma de muestras. Esta etapa corresponde al módulo AFE, de la arquitectura que se propone.

4.1.2 Conversión Análoga/Digital o Toma de Muestras

Corresponde a todo el proceso conversión de señales análogas a digitales por parte de los ADC embebidos en el MCU y a la transferencia de datos del ADC a memoria, realizada por el DMA.

4.1.3 Almacenamiento de muestras:

Lo constituyen las tareas de software, que van acumulando los valores de voltaje y corriente al cuadrado y los productos de voltaje por corriente instantánea

4.1.4 Cálculo de parámetros y acumulación de datos:

Con las muestras almacenadas por la etapa anterior, se procede a realizar los cálculos de voltajes rms, corrientes rms, potencia activa, potencia reactiva, potencia total o aparente y energía consumida. También se calculan los costos de la energía.

4.1.5 Comunicación e interfaces:

Los valores calculados de Voltaje, Corriente, Potencia, Energía y Valores de consumo, son mostrados en un display LCD y transmitidos a través de una interface RS232 hacia una terminal e inalámbricamente, utilizando módulos de radio ZigBee.

La explicación detallada de la implementación de las anteriores etapas se hace en el capítulo 5.

4.2 Plataforma de desarrollo:

La implementación del Smart Meter, del presente proyecto, se lleva a cabo, con el uso de un microcontrolador ARM córtex M4, de la familia SAM4E16E de Atmel Corporation[25] (Hoy Microchip) y sus respectivos periféricos, mediante el uso de una plataforma de desarrollo, la SAM4E-EK [26].

Este MCU, posee las siguientes características importantes:

Procesador ARM córtex M4 con FPU (Floating Point Unit), instrucciones DSP ,120MHz, 128KB RAM, 1MB Flash, RTC de 32 bits con calendario Persa y Gregoriano, 3 Timers de 32 bits 3 canales cada uno, 2 ADC de 12 bits de 16 canales cada uno con secuencia automática de conversión programable, 2 Controladores DMA (PDC, Peripheral DMA Controller), 2 UART, mas de 100 GPIOs.

Como su fabricante lo dice es un procesador ideal para aplicaciones industriales y de manejo de energía.

Este MCU no dispone de 7 ADCs, que se necesitarían para la toma de muestras en un sistema trifásico de 4 hilos y parte del trabajo consiste en tratar de hacer un óptimo uso de los recursos disponibles. En este caso solo se dispone de 2 ADCs.

El Timer es un periférico indispensable en este tipo de desarrollos y para la aplicación específica de este diseño, se requiere que los Timers a usar se puedan sincronizar. El SAM4E16E, tiene esta propiedad. Adicionalmente es importante, que los Timers puedan dar el trigger periódico al ADC, para minimizar intervenciones de la CPU. EL SAM4E16E, permite la programación del trigger periódico de los ADC, desde los timers

El DMA (Direct Memory Acces), es otro periférico indispensable cuando los ADCs, con los cuales se realiza la toma de muestras están embebidos en el MCU. Como la toma de muestras es periódica

e ininterrumpible, se requiere que el proceso de toma de muestras, no tenga intervención de la CPU. Con este MCU, se tiene un DMA de varios canales, que permiten pasar los datos muestreados a la memoria sin intervención de la CPU.

Otro periférico importante es el Real Time Calendar (RTC), el cual permite tener actualizada la fecha y tiempo de la toma de muestras, como programar eventos periódicos.

Para complementar la plataforma de desarrollo SAM4E-EK, se ha diseñado una tarjeta PCB, de acondicionamiento de señal, que se ha llamado Analog Front End (AFE). Esta tarjeta contiene los circuitos necesarios para lograr que las señales de voltaje y corriente, cumplan con los requerimientos del sistema y las especificaciones de los ADC del MCU.

4.3 Arquitectura Propuesta.

Para cumplir con la metodología expuesta de desarrollar el sistema en las etapas previamente mencionadas, se propone una arquitectura basada en módulos o bloques de hardware y software.

El sistema se implementará con dos tarjetas: Una de acondicionamiento de señal (AFE) y la tarjeta de desarrollo SAM4E-EK, basada en el MCU SAM4E16E. Con base en ellas, se han diseñado los módulos de hardware y software necesarios, para lograr el sistema propuesto.

La figura 4.1 ilustra el diagrama general bloques con los módulos de hardware y software del sistema propuesto. La descripción detallada de cada bloque y su tarea asociada se hace en la sección 5. Los módulos de hardware y software, se distinguen con colores azul y naranja respectivamente.

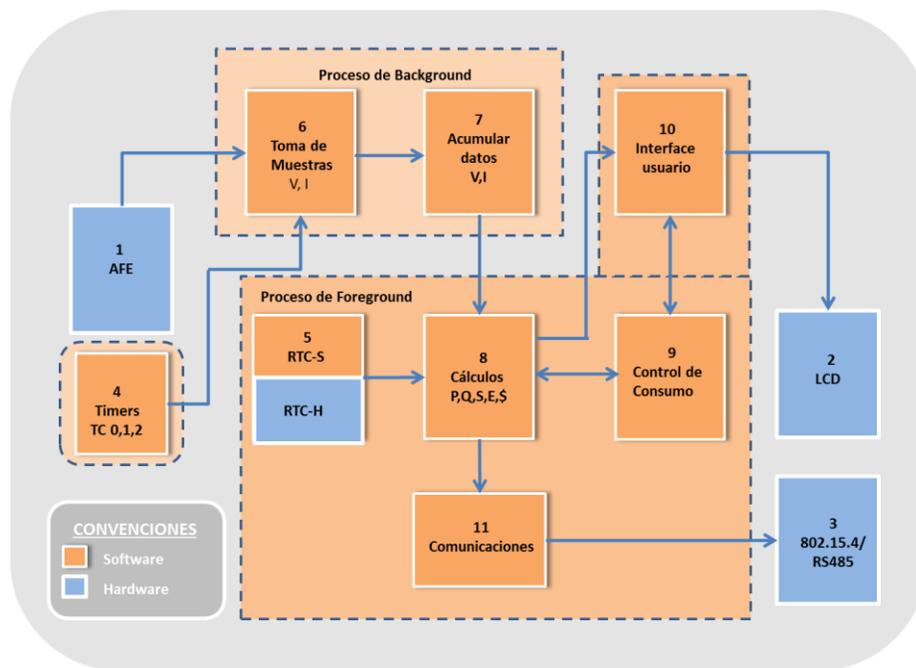


Figura 4.1 Diagrama general de bloques del Smart Meter propuesto

Un aspecto novedoso del presente diseño, es la implementación de un método de toma de muestras de las señales, en buffers pequeños, que pueden ser guardados en memoria RAM. Es decir que en lugar de acumular todos los datos durante un segundo, en memoria flash, se acumulan sub-buffers en RAM, que luego se van guardando para completar el buffer de un segundo. Esto hace que el sistema sea muy eficiente en cuanto al uso de memoria. Este método se explicará en la sección 5.2.1

Con este método, se reduce considerablemente el tamaño de memoria, necesario para almacenar los datos muestreados de corriente y voltaje en cada ciclo, como se explicará en la sección 5.2. La disminución del tamaño de la memoria se aplica, independientemente de donde se estén guardando para su posterior cálculo: Memoria RAM o Memoria Flash

Este método se explicará en la sección 5.2.1

5 DESARROLLO E IMPLEMENTACIÓN

En esta sección se describen en forma detallada, los procedimientos realizados para lograr el sistema propuesto. Se describen tanto los circuitos y componentes que forman cada bloque de hardware con su diseño y cálculo de valores, como una descripción de las tareas de software que se implementan en cada módulo y sus interfaces.

5.1 Implementación del hardware. Etapa 1

El hardware está compuesto por la tarjeta de desarrollo y una tarjeta adicional de acondicionamiento de señal AFE.

La tarjeta de desarrollo SAM4E-EK [6], y la tarjeta de adecuación de señales AFE, se ilustran en la figura 5.1

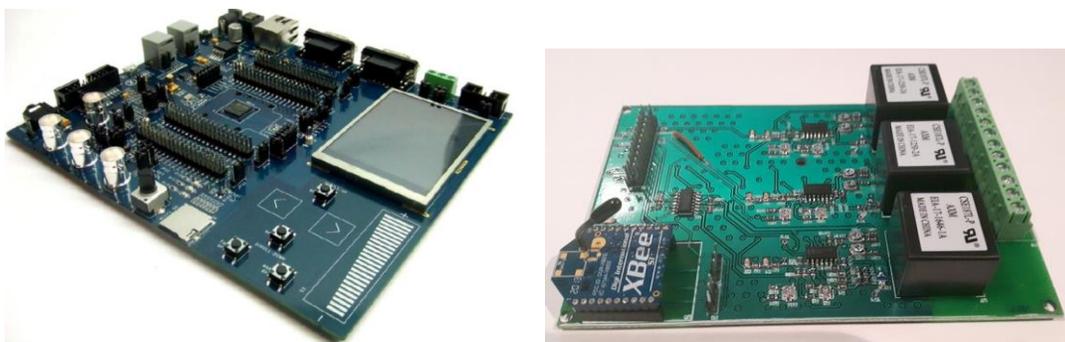


Figura 5.1 Tarjeta de desarrollo SAM 4E-EK [26]. y tarjeta AFE

La tarjeta AFE, incluye los circuitos para adecuar cada una de las señales de entrada, tanto de voltaje como de corriente, es decir el circuito de muestreo compuesto por detector de conexiones, señal de sincronismo, level shifter y filtro antialiasing; También la tarjeta AFE incluye el módulo ZigBee de transmisión inalámbrica, XBee S2 de Digi International, encargado de transmitir los datos hacia un punto de recepción, como puede ser una terminal. El anexo 1 contiene los esquemáticos de diseño del PCB, realizados con ALTIUM 16

A continuación se describen los distintos módulos de hardware del AFE

5.1.1 Muestreo de Voltaje

La entrada análoga al ADC, por especificaciones del microcontrolador, debe tener valores entre 0 V y 3.3V máximo, es decir que se hace necesario un level shifter, para quitarle la componente negativa a la señal AC. Esto se logra sumando un offset de 1.65 V o 1.5 V (según el valor de VREF del ADC) a la señal AC de entrada. Debe aclararse que el ADC se trabaja en modo single ended.

La especificación exige que se pueda medir un voltaje de red de 102-140 Vrms (ver especificaciones del sistema en el capítulo 3), es decir hasta 198 Vmax. Se necesita que para este valor, la entrada al ADC sea máxima de 3.3 V, en modo single ended. Sin embargo en el presente

diseño se deja un margen de seguridad, para evitar recortes en la señal del ADC (clipping) y el valor máximo será de 3.0 VDC, es decir que V_{REF} en el ADC será de 3.0 V.

El circuito de la figura 5.3 ilustra la toma de muestras de voltaje de una de las fases con su respectivo detector de cruce por cero. Se ha diseñado con un transformador de aislamiento por estrictas razones de seguridad de la plataforma de desarrollo, pero la señal también se puede muestrear mediante divisores de voltaje resistivos. El transformador TX, es el 164D10 de Hammond Manufacturing, por ser tipo PCB de pequeño tamaño y voltaje de entrada 110 V/220 V. Tiene una relación $n = 12$ y es tipo PCB.

El voltaje hacia el ADC es reducido a través del divisor formado por las resistencias R_1 y R_2 y el potenciómetro P1. En los cálculos se asume que R_2 incluye el valor del potenciómetro. El potenciómetro P1, calibra el valor máximo de entrada al ADC, para que no exceda los 3.3V, permitiendo que el valor máximo de la señal de salida del transformador sea ajustable, lo cual brinda la posibilidad de tener transformadores de entrada con diferentes relaciones de transformación.

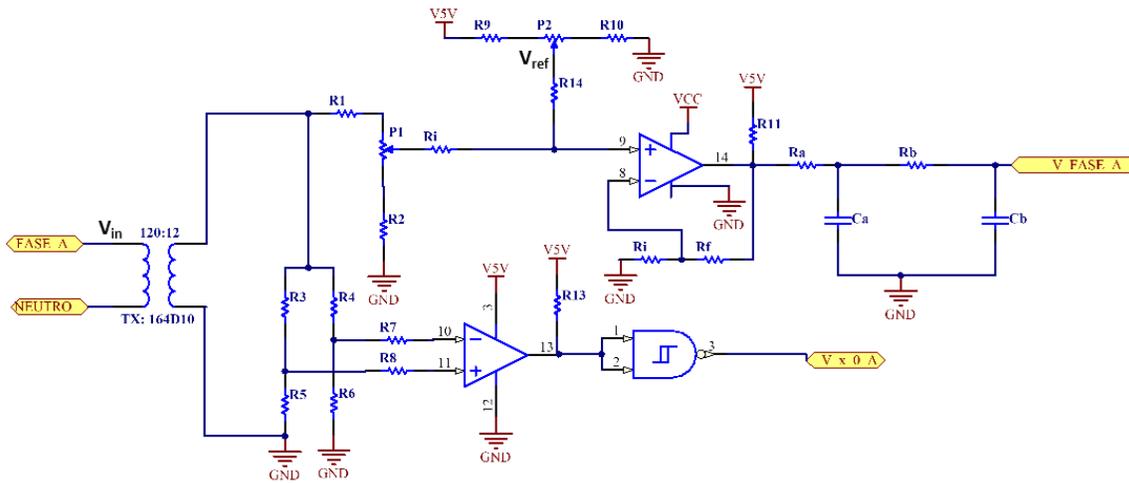


Figura 5.3 Circuito de muestreo de voltaje. Altium - James Zafra

Como el filtro antialiasing, compuesto por $R_a C_a, R_b C_b$, atenúa muy poco la señal por debajo de 1920 Hz, A_{in} estará determinada por la atenuación del divisor resistivo, mas el voltaje de referencia y la ganancia del operacional (Sin embargo en el proceso de calibración, todas las atenuaciones de la señal se tienen en cuenta). De la figura 4.3, la entrada no-inversora al opamp, V_{2+} , está dado por:

$$V_+ = \sqrt{2}V_{in} \left(\frac{1}{n} \right) \frac{R_2}{R_1 + R_2} \frac{R_{14}}{R_i + R_{14}} + V_{ref} \frac{R_i}{R_i + R_{14}} \quad (5.1)$$

El máximo valor de entrada al ADC (3.0 V), ocurre cuando el valor RMS de entrada es de 140 V.

$$3.0V = \frac{1}{2} \left[140\sqrt{2} \left(\frac{1}{n} \right) \frac{R_2}{R_1 + R_2} + 1.5 V \right] \quad (5.1a)$$

La ecuación (5.1 a), conduce a la relación

$$R_1 = 2.7R_2 \quad (5.2)$$

Si seleccionamos $R_2 = 1.2 \text{ k}\Omega \Rightarrow R_1 = 3.3 \text{ k}\Omega$. Como el valor de R_2 es ajustado por el potenciómetro P1, se selecciona en $1.0 \text{ k}\Omega$ en serie con un potenciómetro de $1 \text{ k}\Omega$. Si se combinan las ecuaciones (5.1) y (5.2), teniendo en cuenta la relación de vueltas del transformador y agregando la ganancia del amplificador operacional ($G = \frac{R_f}{R_i}$) podemos obtener la ecuación que nos da el valor de entrada al ADC en función del voltaje de entrada de la red y de la ganancia del operacional.

$$V_{Ain} = \frac{1}{2} \left(V_{in} x \sqrt{2} \left(\frac{1}{12} \right) x \frac{R_2}{2.7R_2 + R_2} + V_{ref} \right) G = 0.016xGxV_{in} + 0.75G$$

$$V_{Ain} = 0.016x V_{in} + 0.75 \quad ; \text{ con } G = 1 \quad (5.3);$$

La ecuación (5.3) nos dice que por cada voltio rms de la red AC de entrada, tendremos $0.016V$ en la entrada analógica del ADC, siempre que la relación del transformador sea $120/10$, $V_{ref}=1.5$ y $G=1$.

La especificación en la medición de voltaje del sistema es del 2.5% , (IEC62053-21 clase 2). Esto significa que a voltaje mínimo de entrada (102 V rms), la variación máxima debe ser de 2.55 Vrms , que al aplicar 5.3, corresponde a una variación de 0.016 V a la entrada del ADC.

La resolución deseada del ADC, es 16 mV/bit , los 0.75 son un Offset que se debe remover, luego:

$$LSB = \frac{V_{REF}}{2^n} = 16 \text{ mV} \quad (5.4)$$

Y como $V_{REF} = 3.0 \text{ V}$ despejando n tendremos:

$$n = \frac{1}{\ln 2} \ln \left(\frac{V_{REF}}{LSB} \right) = \frac{1}{0.69} \ln \left(\frac{3.0}{0.016} \right)$$

$$n = 7.58 \text{ bits} \Rightarrow 8 \text{ bits}$$

Lo anterior indica, que para cumplir con nuestro rango y precisión de voltaje especificados ($100 - 140 \text{ V rms}$) y 2.5% de precisión, necesitamos un ADC de 8 bits reales (ENOB).

5.1.2 Muestreo de Corriente

Con respecto a la corriente, se deben poder medir corrientes de 0.3 A a 30 A . Consideraremos tres formas de medir la corriente:

- Resistencia shunt (R_s):

Consiste en una resistencia R_s de muy bajo valor ($\mu\Omega$, $m\Omega$), que se instala en serie con el circuito a medir, de tal forma que a plena carga, la caída de voltaje sea pequeña y no afecte la operación normal de la carga. Tiene como ventajas, no producir desfase entre la corriente medida y el voltaje

de entrada al ADC, además de su bajo costo y su comportamiento eléctrico lineal. Sus desventajas radican en la necesidad de interrumpir el circuito a medir para poder instalarla, el consumo de potencia, y en que no hay aislamiento entre el circuito de potencia y el de control.

En nuestro caso se debe seleccionar una resistencia que a máxima carga (30 A), produzca un voltaje de 3.0 V, es decir tal que $3.0 V = \sqrt{2} \times 30 A \times R_s$ o $R_s = 70.7 \text{ m}\Omega$. Esto implica un consumo de potencia sobre la resistencia de 13watts, el cual es inaceptable para esta implementación. Se puede seleccionar un valor menor y amplificar la señal pero esto eleva la cantidad de componentes y el costo y por esta razón, adicional a sus desventajas, se desecha esta alternativa.

- Transformador de Corriente CT (Current Transformer):

Este tipo de sensor, consiste en un transformador con una espira en el primario y una relación de vueltas elevada, para que el primario (de una espira) conectado a la carga produzca una corriente pequeña en el secundario (de muchas espiras). En el secundario, se debe conectar una resistencia de carga (R_{burden}) para producir el voltaje deseado. El fabricante recomienda un valor máximo de resistencia. Esta resistencia debe estar siempre conectada, para evitar voltajes elevados que pueden dañar el ADC, el MCU e incluso el mismo CT. La figura 6. Ilustra el circuito equivalente de un CT.

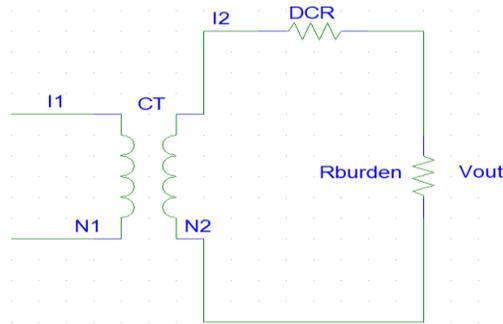


Figura 5.4. Circuito equivalente de un CT

La ecuación, que relaciona el número de vueltas con la corriente de cada devanado es $I_1 N_1 = I_2 N_2$. Dado que $N_2 \gg N_1$, se tendrá una corriente I_2 muy pequeña comparada con I_1 . Para nuestro caso necesitamos medir 30 A en el primario y que esta corriente produzca un voltaje pico máximo en el secundario de 3.0 V, que será la entrada del ADC.

El fabricante como lo ilustra la figura 5.4, normalmente especifica la resistencia serie del secundario (DCR), el valor máximo de voltaje de salida V_{out} y en algunos casos la resistencia Burden. Por ejemplo el modelo CSE187L-P de Triad Magnetics [8], tiene una relación de vueltas de 1:500, un V_{out} máximo de 4.0Vrms a la salida y una DCR de 21 Ω . Esto significa que se tendrá una corriente máxima en el secundario de 60 mA (30A/500). Adicionalmente el fabricante sugiere una resistencia burden $R_b = 60 \Omega$, que permite tener variaciones aproximadas de 110mV/A [8]. Con estos valores y a máxima corriente de 30 A, se produce un voltaje máximo de salida de 3.6 Vrms (60 mA x 60 Ω).

Nosotros seleccionaremos una resistencia de 36Ω , para que a corriente máxima tengamos $3.0V$ máx a la entrada del ADC ($\sqrt{2} \times 60 \text{ mA} \times 36 \Omega = 3.0 \text{ V}$).

El valor de $R_{Lmax} = 36 \Omega < 60 \Omega$, nos garantiza estar en la región lineal de la curva de histéresis del CT para que no se sature. El voltaje total rms, sobre los terminales del CT será de $I_2 \times (DCR + R_{burden}) = 60 \text{ mA} \times (36 \Omega + 21 \Omega) = 3.42 \text{ V}_{rms}$ (véase la figura 4.2), que también está por debajo del límite de la especificación del CT.

La resolución necesaria del ADC, en este caso se calcula con la corriente mínima que se debe medir ($300\mu\text{A}$). Este valor implica que en el secundario del CT se tendrán $0.6\mu\text{A}$ ($300 \mu\text{A}/500$), que corresponden a un voltaje de entrada al ADC de 21.6 mV ($0.6 \mu\text{A} \times 36 \Omega$). Para este valor el número de bits será:

$$n = \frac{1}{0.69} \ln \left(\frac{3.0}{0.0216} \right) = 7.15$$

Que se satisface con un ADC de 8 bits reales o efectivos (ENOB)

- Transformador de corriente tipo Rogowski:

También conocida como bobina de Rogowski, consta de una bobina enrollada en forma helicoidal, pero con núcleo de aire, por cuyo interior se hace pasar el conductor al cual se le desea medir la corriente, como lo ilustra la figura 5.5.

La ley de Faraday, establece que la fuerza electromotriz $V(t)$, inducida por un campo magnético $\Phi(t)$, es proporcional a la tasa de cambio de éste con respecto al tiempo

$$V(t) = \frac{d\Phi}{dt} = A \frac{dB}{dt} = \mu A \frac{dH}{dt}$$

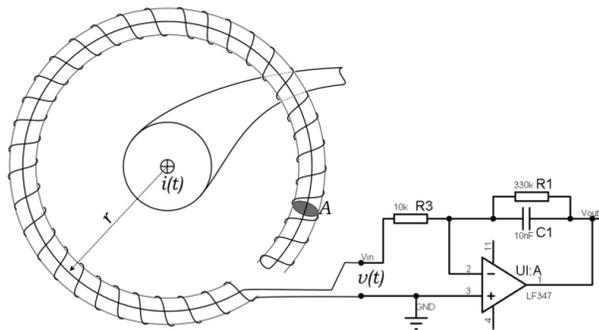


Figura 5.5. Bobina de Rogowski [29].

Siendo A el área del campo inducido y $N=1$ (Una espira). De la ley de Ampere que asocia el campo magnético H a la corriente \dot{i} que lo produce, obtenemos:

$Hl = Ni$, que al reemplazar nos da

$$V(t) = \mu A \frac{N di}{l dt}$$

De esta última ecuación deducimos que el voltaje inducido, por la corriente i , en los terminales 1, 2 es proporcional a la derivada de la corriente. Es decir que si se desea tener a la entrada del ADC un voltaje que sea proporcional a la corriente de entrada se tiene que agregar un integrador al circuito.

Este tipo de sensor tiene varias ventajas: Es bastante lineal y no dependiente de la saturación de materiales magnéticos, tiene una respuesta muy rápida debido al bajo valor de inductancia de la bobina y aunque ella misma no produce desfase de la corriente de entrada, este se produce al tener que integrar la salida y por tanto también necesita un circuito compensador de fase. El sensor permite medir corrientes muy altas, tiene poca variación con la temperatura y debería ser de bajo costo por no tener material magnético, entre otras características. Sin embargo, aún no son muy comerciales como los CTs y esto hace que su precio sea alto para un prototipo. Por su costo y circuitos adicionales necesarios no se ha seleccionado.

- **Selección del Sensor de Corriente :** Para el presente proyecto se ha hecho la selección del CT CSE187L-P de Triad Magnetics [8]. Este CT posee las siguientes especificaciones:

Especificación	CT CSE187L-P
I _{max}	30 A
Relación de Vueltas	500
Salida típica	110 mV/Amp
Máximo Voltaje de salida	4.0 V rms
DCR Primaria	250 $\mu\Omega$
DCR Secundaria	21 Ω max
R Burden sugerida	60 Ω

Tabla 4.1 Especificaciones del transformador de corriente

El circuito para el muestreo de corriente para una fase, se ilustra en la figura 5.6.

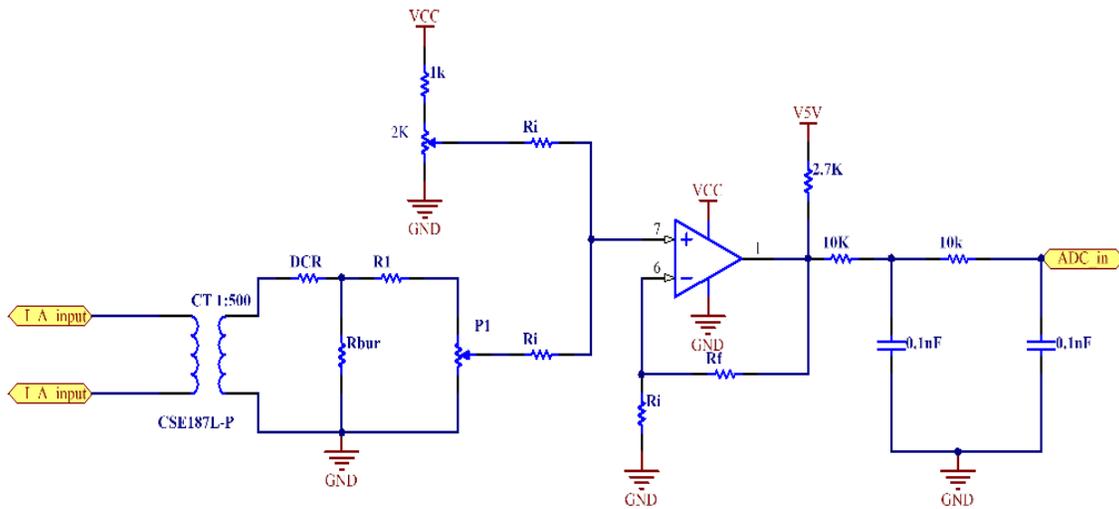


Figura 5.6. Circuito de Muestreo de Corriente. Altium - James Zafra

5.1.3 Filtro Anti – Aliasing

Las resistencias R_a , R_b y los capacitores C_a , C_b de la figura 5.1, forman el denominado filtro pasivo anti-aliasing de segundo orden de entrada al ADC. Este filtro se encarga de eliminar las frecuencias no deseadas, que estén por encima de la frecuencia mínima que exige el teorema de Nyquist, es decir F_s . Se trata de un filtro pasa bajos en cascada. La ecuación que relaciona el voltaje V_o de salida del operacional con el voltaje V_i de entrada al ADC, es

$$\frac{V_o}{V_i} = H(s) = \frac{1}{1 + SRC} \quad (5.5) \text{ siendo } S = j\omega c$$

Esta es la función de transferencia de cada etapa del filtro. A frecuencia corte, el filtro tiene una ganancia de $-3dB$ o amplitud de 0.707, que equivale a tener una potencia de 0.5 veces la potencia de la señal. Es decir a f_c , $\frac{P_o}{P_i} = 0.5$ y $GdB = 10 \log\left(\frac{P_o}{P_i}\right) = 10 \log(0.5) = -3 \text{ dB}$

Es recomendable que el muestreo, permita capturar el armónico 32nd. porque, para un Smart Meter tipo industrial, el estándar exige que esto se realice y nuestro diseño puede ser escalable en un futuro a este tipo de Smart Meter. Esto implica que se tomen como mínimo 64 muestras por ciclo, para cumplir con el teorema de Nyquist ($f_s = 2 \times 32 \times 60$). A la vez, es deseable que el número de muestras por ciclo de la señal sea múltiplo de 2^n , para que si se desea calcular el contenido de armónicos se pueda implementar fácilmente la FFT (Fast Fourier Transform). 64 es 2^6 , que cumple esta condición.

La frecuencia de muestreo será, de acuerdo al teorema de Nyquist,

$$f_{\text{sampling}} = 2 \times f_{\text{armónico}} \times f_{\text{señal}} = 2 \times 32 \times 60 = 3840 \text{ Hz} .$$

Deseamos que a la mitad de esta frecuencia, se tenga la atenuación de 3dB

$$A_1 = \left| \frac{V_o}{V_i} \right|_{1920} = \left| \frac{1}{1 + SRC} \right| = \frac{1}{\sqrt{1 + \omega_1^2 R^2 C^2}} = \frac{1}{\sqrt{2}} \quad (5.6)$$

$$\omega_1 RC = 2\pi f_1 RC = 1 \quad (5.7)$$

Al reemplazar en (5.7) el valor de $f_{3db} = 1.920 \text{ Hz}$ obtenemos la ecuación que relaciona a R con C.

$$C_a = \frac{8.29 \times 10^{-5}}{R_a} \quad (5.8)$$

Si se hace $R_a = 10 \text{ k}\Omega \Rightarrow C_a = 0.83 \text{ nF}$; Si $C_a = 1 \text{ nF}$, $R = 8.2 \text{ k}\Omega$. Seleccionando valores

Para mejorar la atenuación, se pueden colocar dos filtros en cascada (ver figura 4.4), en cuyo caso la atenuación será el producto de las atenuaciones de cada filtro.

La tabla 4.1 muestra algunos valores posibles de R y C para el filtro con $f_{-3dB} = 1920 \text{ Hz}$

f	R	C
1920Hz	1 kΩ	0.1uF
	2 kΩ	40 nF
	8.2 kΩ	10 nF
	10 kΩ	10 nF

Tabla 5.1 Valores de f, R y C para el filtro pasa-bajos – RC a 1920 Hz

5.1.4 Muestreo y Conversión.

El circuito interno de muestreo de la señal analógica de entrada al ADC, conocido como S&H (Sample and Hold), se ilustra en la figura 5.7 y está formado por la red RC interna, compuesta por R_i y C_i , más un switch electrónico, normalmente un mosfet (no mostrado.).

Este circuito es el que determina, el tiempo de muestreo de la señal de entrada, también conocido como t_{SAMPLE} o $t_{TRACKING}$ (Se utilizará cualquiera de los nombres $t_{TRACKING}$ o, t_{SAMPLE}).

El tiempo de muestreo $t_{TRACKING}$, es afectado por el valor de la impedancia que ve el ADC, es decir por el valor de impedancia del circuito externo de muestreo (incluido el filtro). Su diseño es importante, porque sus componentes afectan directamente el tiempo total real de la toma de la muestra. No se puede cumplir con determinada velocidad de muestreo, simplemente con las especificaciones de frecuencia del ADC, también es necesario que los componentes de muestreo (R_S y C_S), tengan los rangos de valor necesarios para garantizar esa frecuencia. Hay una relación directa entre la impedancia de entrada al ADC y su frecuencia máxima de muestreo como se verá más adelante.

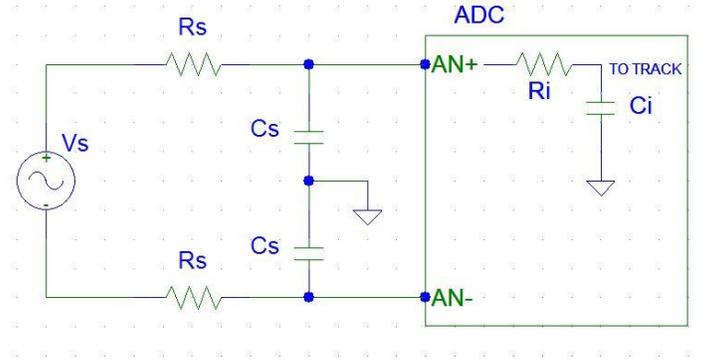


Figura 5.7. Circuito de muestreo de entrada del ADC

El condensador interno del ADC, C_i , tiene un tiempo de carga que depende tanto de la resistencia interna R_i del ADC, como de la resistencia R_s , la cual normalmente está determinada por el diseño del filtro Antialiasing. La ecuación que determina el voltaje sobre el condensador de muestreo C_i , es

$$V_c = V_i \left(1 - e^{-\frac{t}{R_T C_i}} \right) \quad (5.9)$$

Con $R_T = R_i + R_s$. Despejando para hallar el tiempo de muestreo tendremos:

$$e^{-\frac{t}{R_T C_i}} = 1 - \left(\frac{V_c}{V_i} \right)$$

$$-\frac{t}{R_T C_i} = \ln \left(1 - \frac{V_c}{V_i} \right)$$

$$R_T C_i = -\frac{t}{\ln \left(1 - \frac{V_c}{V_i} \right)}$$

$$(R_i + R_s) C_i = -\frac{t_{tracking}}{\ln \left(1 - \frac{V_c}{V_i} \right)} \quad (5.10)$$

Ahora debemos encontrar una ecuación, que nos dé el tiempo de carga de C_i en función de R_i, R_s, f_{ADC} y n , que son parámetros conocidos, siendo R_i la resistencia interna del ADC, R_s la impedancia de la fuente vista por el ADC, $f_{ADC} = \frac{1}{T_{ADC}}$ la frecuencia del reloj del ADC y n el número de bits del ADC.

El cambio más extremo de carga del capacitor C_i , ocurre cuando el voltaje de entrada al ADC, cambia a su valor máximo FSR(Full Scale Range) [27], es decir cuando $V_i = V_{REF}$, en cuyo caso, C_i debe cargarse a $V_c = FSR$. Si reemplazamos estos valores y el de FSR: $FSR = V_{REF} \left(1 - \frac{1}{2^N} \right)$, en la ecuación (5.10), tendremos:

$$(R_i + R_s)C_i = - \frac{t_{tracking}}{\ln\left(1 - \frac{FSR}{V_{REF}}\right)} = - \frac{t_{tracking}}{\ln\left(1 - \frac{V_{REF}\left(1 - \frac{1}{2^N}\right)}{V_{REF}}\right)}$$

$$(R_i + R_s)C_i = - \frac{t_{TRACKING}}{\ln\left(\frac{1}{2^N}\right)}$$

Que, conduce a la expresión general para el **tiempo mínimo** de muestreo $t_{tracking}$:

$$t_{tracking} = (R_i + R_s)C_i \ln(2^N) \quad (5.11)$$

Para un ADC con N=12 bits (que es nuestro caso), el $t_{tracking}$ mínimo será:

$$t_{tracking} = (R_i + R_s)C_i \ln(2^{12})$$

Si en la expresión anterior, el valor de C_i se expresa en pF, y R_i , R_s las expresamos en k Ω , tendremos la expresión general para el tiempo mínimo de muestreo en nanosegundos para un ADC de 12 bits.

$$t_{tracking}(nS) = 8.32(R_i + R_s)C_i \quad (5.12)$$

$$t_{tracking} = \mathbf{8.32R_iC_i} + \mathbf{8.32R_sC_i} \quad (5.12A)$$

5.1.5 Selección del ADC.

Las señales de voltaje y corriente, exigen un ADC de 8 bits de resolución real y frecuencia conversión mínima de 8 kHz que corresponde al doble de la máxima tasa de muestreo. El ADC del MCU, seleccionado para este proyecto, el Atmel SAM4E16E [25], satisface muy bien estos requerimientos y por ello dedicaremos unas líneas al comportamiento de su circuito de muestreo.

El manual del SAM4E16E, de Atmel en la sección 46.7.3.1 nos da la siguiente ecuación para calcular el $t_{tracking}$ en función de la impedancia de entrada, con $t_{tracking}$ en nanosegundos.

$$t_{tracking} = 0.054Z_{source} + 205 \quad (5.13)$$

Podemos hallar los valores teóricos de C_i , y R_i , para el SAM4E16E comparando cada término de las ecuaciones (5.12A) y (5.13)

$$0.054xR_sx10^{-9}seg = 8.32 R_sC_i$$

$$C_i \approx 6.5 pF$$

$$205x10^{-9} = 8.32R_iC_i$$

Al reemplazar el valor de C_i , obtenemos el valor de R_i :

$$R_i \approx 3.8 k\Omega$$

Existe un tiempo mínimo de muestreo, que se debe cumplir y está especificado por el fabricante. En nuestro caso para el SAM4E de Atmel, las características eléctricas del ADC,[9], especifican un tiempo mínimo de S&H de 15 periodos del ADC. Si se desea satisfacer ese tiempo, la impedancia de la fuente (Z_{source}), tendrá un valor máximo que debe satisfacer la desigualdad

$$0.054Z_{source} + 205 \leq t_{tracking} \quad (5.14)$$

Al insertar en la ecuación (5.14), el valor de $R_S = Z_{source}$ (obtenido del filtro Antialiasing y equivalente a R_S en los circuitos de las figura 5.7), se obtiene el $t_{TRACKING}$. Si el valor obtenido es menor que $15t_{ADC} = 15/f_{ADC}$, el valor que se debe tomar para $t_{TRACKING}$ será $15t_{ADC}$. Por otro lado, si el $t_{TRACKING}$ calculado con (5.14) es mayor que $15t_{ADC}$, se toma el $t_{TRACKING}$, pero se debe tener en cuenta que al aumentar el $t_{TRACKING}$, se disminuye la frecuencia de muestreo F_s .

El ADC va a trabajar a $f_{ADC} = 6\text{MHz}$. En este caso $t_{TRACKING}$, será $15t_{ADC} = 15 \times 166.66 \text{ nS} = 2.500 \text{ nS}$ y al reemplazar en (5.14) obtenemos el valor máximo de la impedancia de entrada:

$$Z_{source(max)} \leq 42.5 \text{ k}\Omega \quad (5.15)$$

Esto significa que una impedancia serie de la fuente menor de $42.5 \text{ k}\Omega$, permitirá un $t_{TRACKING} = 15t_{ADC}$, pero una impedancia mayor a $42.5 \text{ k}\Omega$, haría disminuir la frecuencia de muestreo.

5.1.6 Detector de conexión de fases

Un aspecto importante a tener en cuenta, antes de iniciar la toma de muestras, es comprobar que las fases estén conectadas en un orden específico, para que los cálculos de potencia, se realicen con voltajes y corrientes de la misma fase. El sistema implementa una tarea de software que se encarga de verificar que las conexiones de entrada, de las diferentes señales de voltaje, se hagan de forma correcta. Para ello se detectan las interrupciones generadas por cada una de las fases en su cruce por cero. Debe aclararse, que no se trata de un detector de secuencia de fases.

La figura 5.8, ilustra las señales de sincronismo de las tres fases. Con base en ellas, la tarea de determina si las fases están conectadas correctamente.

Se asume que la bornera o conector de entrada, debe tener las conexiones de las fases como lo ilustran las líneas sólidas de la figura 5.8. Es decir, fase A conectada a borne A, fase B conectada a borne B y fase C conectada a borne C.

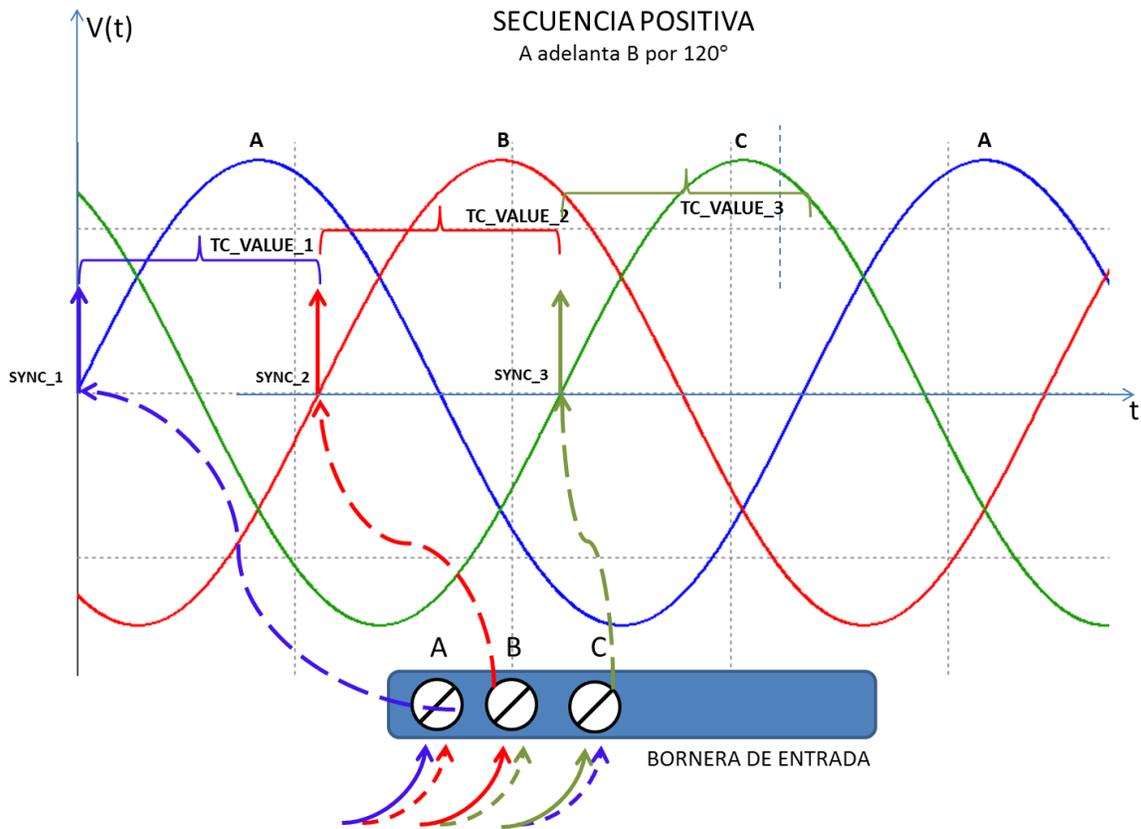


Figura 5.8. Ilustración de detección de orden de fases.

En una secuencia de fases positiva (ABC), si se toma la fase A como referencia (Ver figuras 5.8 y 5.9), la siguiente fase en aparecer será la fase B después de 120° y luego la fase C a 240° . En secuencia positiva, únicamente son posibles las combinaciones ABC, BCA y CAB.

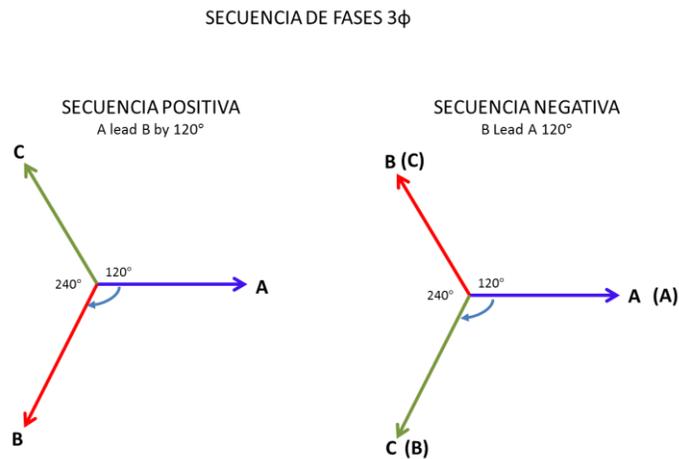


Figura 5.9. Secuencia de fases Positiva y negativa

La implementación consiste en detectar cual de las fases conectadas en la bornera de entrada (A, B, C) genera la primera interrupción y cual la segunda. Hecha esta detección, el sistema avisa a través del display LCD como están las conexiones, para que se haga la corrección. Cualquiera de las tres fases puede generar la primera interrupción, porque ello depende del instante en el cual se inicialice todo el sistema. La toma de muestras solo se puede iniciar, si las conexiones en la bornera corresponden a ABC. El sistema bloquea cualquier otra combinación

Supongamos que las fases en la bornera están conectadas como BCA (Fase B al borne A, fase C al borne B y fase A al borne C), como lo indican las líneas puenteadas en la figura 5.8 y que la primera interrupción la genera la señal SYNC2 (Es decir la fase C conectada al borne B). El sistema no sabe cual es la fase conectada al borne B, pero sabe que la secuencia es ABC. Inmediatamente se asigna al borne B, el distintivo de fase A y se espera la segunda interrupción, que el sistema sabe, debe ser la fase B. En este caso la segunda interrupción la producirá, la señal SYNC1 conectada al borne A. El sistema informará en el display que se tiene la conexión BAC (no confundir con secuencia negativa), para que el usuario haga el cambio a ABC.

5.1.7 Detector de cruce por cero de Voltaje

Las mediciones de voltaje y corriente deben iniciar de forma sincrónica. Esto se logra con un detector de cruce por cero de la señal de voltaje de la fase A, cuyo circuito se ilustra en la figura 5.3. Esta parte del circuito también se utiliza para medir la frecuencia de la línea. El sistema únicamente empieza a tomar muestras, una vez recibida la señal de sincronismo. Se implementan detectores de cruce por cero para las tres fases y para las muestras de corriente.

La señal producida a la salida de la compuerta NAND (figura 5.3), es llevada a uno de los GPIOs (General Purpose Input Output port) del MCU, disponibles en la tarjeta de desarrollo SAM4E-EK, para que genere una interrupción en él, con el borde de bajada.

Las mediciones comienzan, con la detección de cruce por cero de voltaje de la fase A

5.1.8 Módulo de transmisión ZigBee.

El sistema propuesto, exige la transmisión local (In Home) de datos de forma inalámbrica, utilizando el estándar 802.15.4. La utilización del estándar ZigBee (Implementado sobre el estándar 802.15.4), se hace porque es ampliamente utilizado en todo el mundo, tanto por fabricantes de dispositivos de monitoreo de energía, como por empresas prestadoras de servicio (utilities), para medición y transmisión de datos de Energía [12, 21].

Su amplio uso obedece a 5 características fundamentales que posee este estándar [12, 21] :

- ✓ Baja Tasa de transmisión de datos: La máxima tasa de transmisión de datos del estándar ZigBee es de 250Kbps.
- ✓ Orientado a dispositivos alimentados por batería: [12][21]. ZigBee, se utiliza en dispositivos de monitoreo de pacientes , dispositivos de medición de consumo de energía,

monitoreo de grandes estructuras de edificaciones, etc. En este tipo de aplicaciones, se requiere que el dispositivo opere de forma continua, por largos periodos de tiempo.

- ✓ Desarrollado para aplicaciones de bajo costo, bajo consumo y baja tasa de transmisión de datos: En muchas aplicaciones, el módulo ZigBee permanece en estado inactivo la mayor parte del tiempo y solo despierta, para transmitir sus datos. El Duty Cycle de uso del módulo ZigBee puede ser menor al 1% (como se verá más adelante), lo cual asegura una larga duración de la vida útil de las baterías.
- ✓ Formación de redes con gran cantidad de nodos. Teóricamente se pueden formar hasta 65536 nodos en una red y además se pueden formar redes de redes.

Un Smart Meter debe funcionar de forma continua durante tiempos superiores a 5 años, alimentado por batería, porque las mediciones no pueden detenerse ante una falla en la alimentación del dispositivo, es decir que debe ser de bajo consumo[4,5]. Adicionalmente debe tener capacidad de transmisión, para enviar los datos de medición bien sea hacia un dispositivo donde se muestren en el hogar (In Home Display), o hacia otro dispositivo de recolección y re transmisión.

La implementación propuesta, transmite datos de consumo de potencia, energía y costo acumulado, de forma inalámbrica, hacia otro dispositivo, para su recolección o display, utilizando el estándar 802.15.4. Para lograrlo, se utilizan los módulos Xbee S2 o Xbee PRO, International variant (cualquier pareja o combinación de estos módulos funciona) de la marca Digi International®.

La transmisión de datos de medición, se realiza desde la tarjeta de desarrollo SAM4E-EK vía UART, hacia un módulo Xbee S2, el cual es instalado en la tarjeta AFE. Este módulo transmite vía radio los paquetes, a un segundo módulo, conectado a una terminal serial para que el usuario los pueda visualizar. El esquema de transmisión se ilustra en la figura 5.10

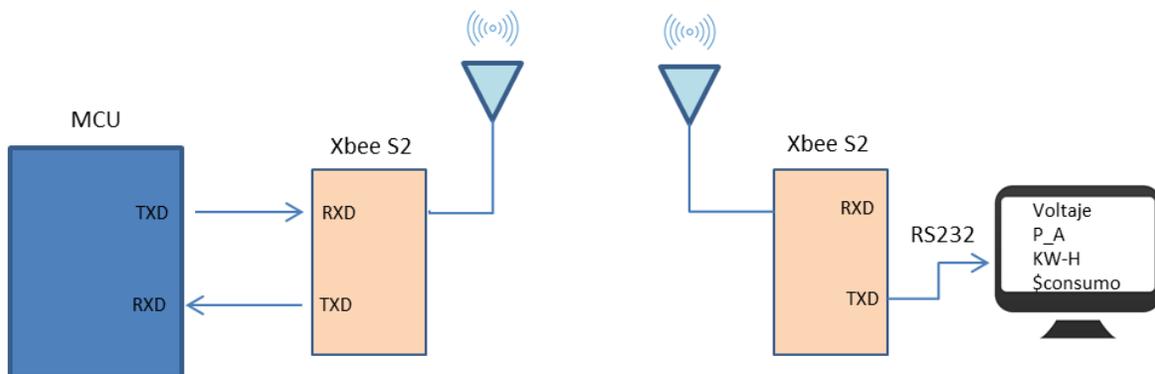


Figura 5.10. Configuración para transmisión de datos por 802.15.4

Los módulos Xbee y Xbee PRO, son compatibles entre sí pin y se pueden combinar para armar los dos puntos de transmisión. La Tabla 5.2 muestra las características principales de estos dispositivos

Especificación	Xbee S1/S2	Xbee PRO
Frecuencia operación	2.4 GHz	2.4 GHz
Alcance: Indoor/Urbano	30 m	90/60 m
Alcance Outdoor	60 m	750 m
Potencia TX	1 mW	10 mW
Sensibilidad RX	-92 dBm	-100 dBm
RF data rate	250 Kbps	250 Kbps
Voltaje	3.3 V	3.3 V
Interface serial	1200-250 Kbps	1200-250 Kbps
Número de canales	16	12

Tabla 5.2 : Especificaciones Eléctricas de los Módulos Xbee S/S2 y Xbee PRO

5.2 Implementación de Módulos de Software

La implementación del software, del sistema propuesto, se ha realizado en lenguaje ANSI C, utilizando como ambiente de desarrollo el programa ATMEL STUDIO 7.0. El diseño corresponde al desarrollo de cada uno de los módulos propuestos en el diagrama de bloques de la figura 4.1 y se ha elaborado de forma modular (Cada bloque y cada periférico tienen un módulo de software), eliminando la programación TOP/DOWN. Esto facilita la comprensión del software por parte de un tercero, las modificaciones o actualizaciones que se deseen en un futuro y la portabilidad del código.

Se ha desarrollado la programación de cada módulo tratando de hacerlo lo más independiente posible de los demás, de tal forma que su código pueda aplicarse en otras implementaciones, diseñando las respectivas interfaces de comunicación entre módulos.

Cada módulo de software involucra diferentes periféricos del MCU, necesarios para su implementación. Para mantener uniformidad en la programación, que facilite su lectura y comprensión y que permitan su uso en diseños posteriores relacionados, todos los módulos se han diseñado con las siguientes reglas establecidas en este diseño:

- ✓ Cada módulo de software, tiene su nombre en mayúsculas y debe referirse en lo posible, al periférico que lo implementa. Ejemplo ADC.c, es el módulo de los ADC
- ✓ Cada módulo contiene un archivo header o **.h** con el mismo nombre del módulo en C y con las siguientes características: definición de variables, definición de apuntadores, una estructura con parámetros de configuración, una estructura con datos modificables, definición de funciones.
- ✓ Todas las variables y defines (**#define**), se nombran al comienzo del archivo .h o .c, después de los **#defines** y con letra mayúscula.
- ✓ Los apuntadores se definen con letras minúsculas. Los apuntadores se nombran, precedidos por la letra p (pointer) y su nombre trata de especificar claramente, la estructura, función u objeto al cual apunta.. Ejemplo: un apuntador a la estructura con registros del ADC, debería tener el nombre: *p_adc , o *padc.
- ✓ Las funciones, se nombran con letras minúsculas y contienen el nombre del periférico respectivo. Ejemplo: **void** configuracion_adc (**void**).
- ✓ Cada módulo contiene las siguientes funciones con sus parámetros acordes al periférico: Función de configuración o Setup, Función de inicialización, Función de configuración de interrupción , Función de interface o llamado, otras.
- ✓ Las funciones, en lo posible contienen un máximo de tres (3) parámetros.
- ✓ Ninguna función contiene estructuras o arreglos como uno de sus parámetros. Se prefieren los apuntadores.
- ✓ Las funciones tienen un comentario previo, para saber cual es su objetivo
- ✓ Las variable, defines y apuntadores también contienen comentarios

Varias implementaciones de Smart Meter [16,17, 18], se realizan, diseñando el software, mediante dos procesos, llamados background y foreground. El proceso de background, se encarga de recibir

los datos de voltaje y corriente de los ADC y guardarlos en buffers de memoria, para luego pasarlos al proceso de foreground, que se encarga de realizar los cálculos necesarios para obtener las mediciones deseadas. El presente proyecto también adopta esta metodología. A continuación se describen en detalle estos dos procesos y su implementación.

5.2.1 Proceso de Background

Este proceso, es el encargado de tomar las muestras de voltaje y corriente, producidas por cada uno de los ADC y sus canales e ir acumulándolas durante 1 segundo para pasarlas al proceso de foreground. En el presente diseño, está formado por los módulos de Toma de Muestras ADC-DMA y Lectura de datos V, I ilustrados en el diagrama de bloques de la figura 4.1. El diagrama de flujo de la figura 5.9, ilustra el proceso. La figura 5.11 ilustra este proceso

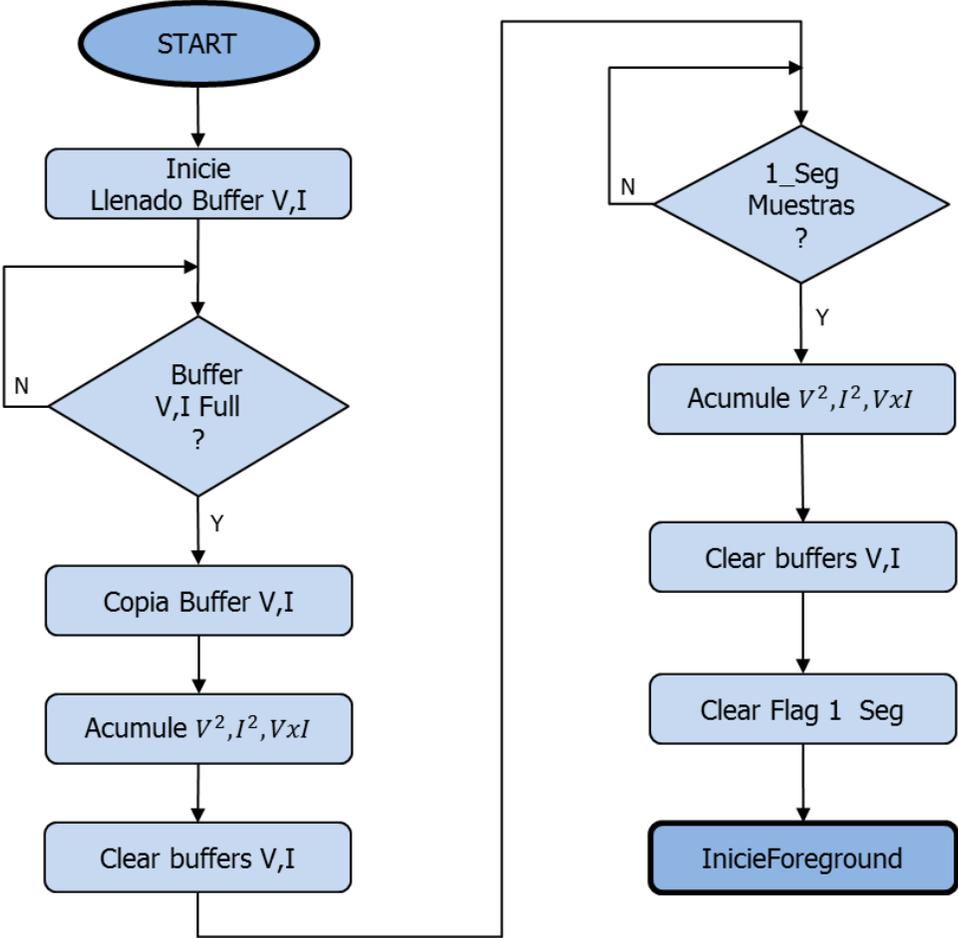


Figura 5.11. Diagrama de flujo del proceso de background implementado

5.2.1.2 Sub-Buffering. Método propuesto. Etapa 2

El proceso de Background, debe funcionar continuamente, porque las mediciones de consumo de Potencia y Energía, nunca se detienen. Además este proceso debe funcionar, sin ocupar tiempo de la CPU. Para lograr lo anterior, es indispensable, que la toma de datos por parte de los ADC, se realice continua y periódicamente, sin que su reinicio lo realice el software. Esto se logra, obligando a que la señal de trigger de los ADCs, las generen dos Timers sincronizados y haciendo que los datos producidos por el cada ADC, se pasen a la memoria, a través del DMA. el cual genera una interrupción a la CPU, cuando su Buffer (Previamente definido en el programa), esté lleno.

Cuando se toman todas las muestras durante 1 segundo y se van guardando en memoria flash, para realizar los cálculos con ellas, el tamaño de la memoria flash es considerable. Incluso si se tomaran muestras únicamente durante varios ciclos, hay que tener suficiente espacio en memoria flash, para acumular los datos.

Supongamos que se desea muestrear un sistema trifásico de 4 hilos, con 256 muestras por ciclo y que se implementa en MCU de 16 bits. Si se acumulan los datos durante 1 segundo, implicaría tener una memoria flash, con un mínimo espacio para datos de:

$256 \text{ muestras/ciclo} \times 7 \text{ canales} \times 60 \text{ ciclos} \times 2 \text{ bytes/muestra} = 215.040 \Rightarrow 256 \text{ KB}$. Si se adiciona el espacio de memoria ocupado por el programa, el MCU debe disponer aproximadamente de 320 KB a 384 KB. El costo de un MCU, se incrementa al aumentar la memoria flash y mantener las demás características.

Aún si se muestrean varios ciclos, la memoria es apreciable. Si acumularan datos cada 4 ciclos en el ejemplo anterior, se necesitaría una memoria de 16KB, para datos. El costo que hay que pagar por ello es el mayor precio del MCU a medida que su memoria flash aumenta

Por ejemplo, en Mouser Electronics (www.mouser.com), para el día 10/11/2017, dos MCU de Texas Instruments, con características muy similares, pero uno con 64 KB de Flash y el otro con 256KB de flash tenían los siguientes costos:

MSP430FR5969IRGZR(64 KB) \Rightarrow US\$4.78 y el MSP430FR5969IRGZR(256 KB) \Rightarrow US\$6.81

Una de las contribuciones más importantes del presente trabajo, es la implementación de un método de captura de muestras, más eficiente en el uso de memoria, a los métodos comúnmente implementados en la mayoría de los trabajos, existentes y previamente referenciados, que redonda en menor costo y posible uso de un MCU muy sencillo para implementación.

El presente diseño, **implementa un método que se ha denominado sub-buffering** que consiste en tomar las muestras de voltaje y corriente, durante periodos, menores a 1 ciclo de la fundamental que se desea medir (sub_buffers) y que sean sub-múltiplos del número de muestras por ciclo y divisibles por 4 y con estos datos realizar los cálculos que necesita el proceso de background. Adicionalmente, se pueden tener copias del buffer, en memoria RAM y memoria flash.

De esta forma, la memoria ocupada para guardar los datos se disminuye considerablemente y como se tiene una copia en memoria RAM, los cálculos son más rápidos. La figura 5.12 ilustra el proceso, para el caso de las 4 muestras de corriente..

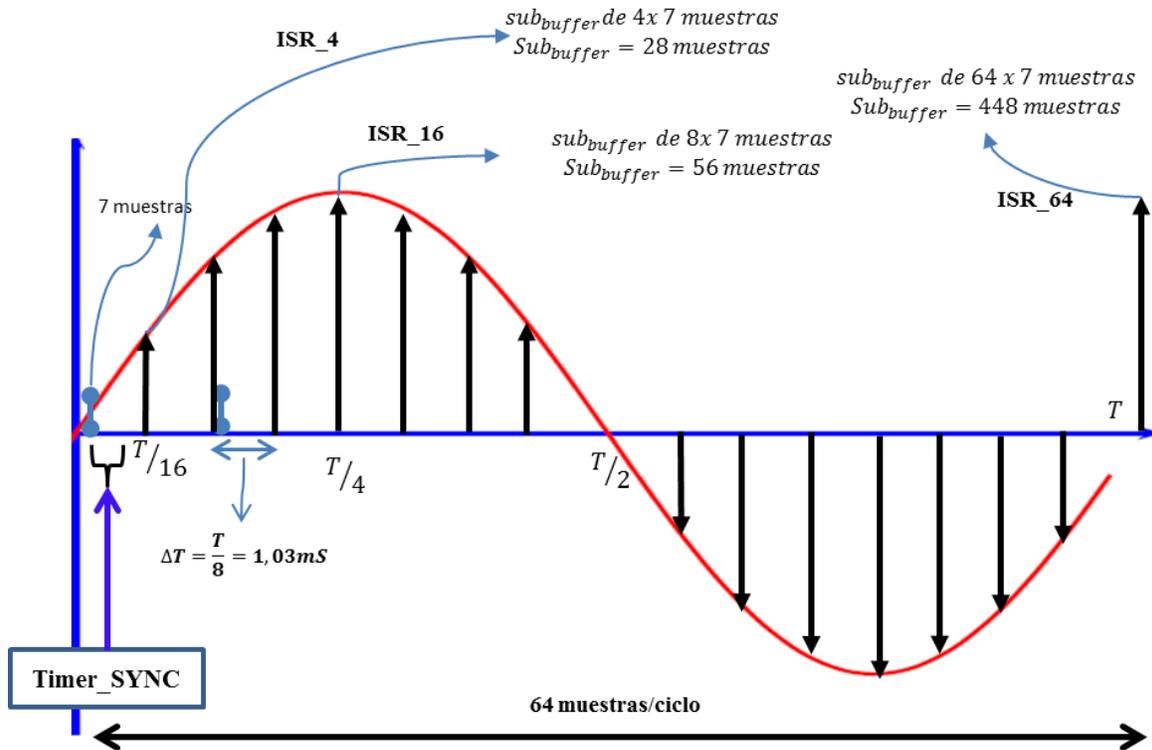


Figura 5.12. Método de muestreo Sub_buffering. Muestras de corriente

La figura 5.12 ilustra los posibles sub-buffers, que se pueden implementar, es decir de 4, 8, 12, 16, etc, muestras. En nuestro caso, se toman 64 muestras por ciclo, luego el sub-buffer más pequeño es de 4 muestras. Como son 4 muestras de corriente correspondientes a las tres fases y el neutro y 3 de voltaje, con cada muestra ocupando un registro de 32 bits se tendrá un espacio en memoria de $4 \times 7 \times 4 = 112$ Bytes de memoria RAM y su respectiva copia en RAM y en flash. Es decir 214 Bytes en RAM y 112 Bytes en flash. Un espacio en memoria bastante pequeño, que muestra la eficacia del método

Una vez se toman las 4 muestras de cada señal, se genera una interrupción, para que el sistema, guarde una copia e inicie los cálculos respectivos. Este es un subproceso dentro del proceso de background.

El precio de este método, está en el límite de tiempo que tiene la CPU, para copiar los datos y realizar los cálculos del proceso de background. Como el proceso de background debe realizar operaciones de acumulado y productos, el tiempo máximo para estas operaciones es de 1.03 mS.

Por que 4 muestras es el tamaño mínimo del buffer?. Porque, si se desea en un futuro implementar la medición de armónicos, es importante poder implementar la FFT.

Debido a este método nuestro proceso de Background, utiliza una rutina de interrupción adicional que hemos llamado Buffer datos V,I. como se muestra en la figura 5.11

5.2.1.1 Módulo de Toma de Muestras y Timers. Etapa 2

Una de las características esenciales, que debe satisfacer un Smart Meter, implementado con un MCU, es que la toma de muestras, la realice un periférico (interno o externo) con el máximo de independencia respecto a la CPU y que esta reciba los datos muestreados mediante una rutina de interrupción del periférico o a través de cualquier otro mecanismo de intercomunicación.

La toma de muestras debe ser sincrónica, tanto en el inicio del muestreo como en la periodicidad de las mismas. Es decir que el sistema debe empezar a tomar las muestras al inicio de cada ciclo de la señal de voltaje de la fase A y repetir el proceso, dependiendo del número de muestras por ciclo especificado. La especificación de este Smart Meter, exige la toma de 3840 muestras de voltaje y corriente en las tres fases y el neutro, por cada ciclo de 60 Hz.

Es decir que para cada fase, necesitamos tomar una muestra de voltaje y corriente, cada $260.417 \mu S$ ($1/3840$). Como únicamente se dispone de 2 ADCs, la metodología de muestreo empleada consiste en utilizar 3 canales de un ADC para tomar las muestras de Voltaje de las tres fases y 4 canales del otro ADC, para las tomas de corriente. El trigger de cada ADC es configurable y permite que se haga con la señal de un timer, es decir sin ocupar tiempo de la CPU.

Cada ADC, recibe una señal de sincronismo de un timer, para producir el inicio de conversión de los tres o cuatro canales como lo ilustra la figura 5.12. Los dos Timers que producen estas señales, inician su conteo de forma sincrónica, para garantizar que las señales de V e I, son simultáneas.

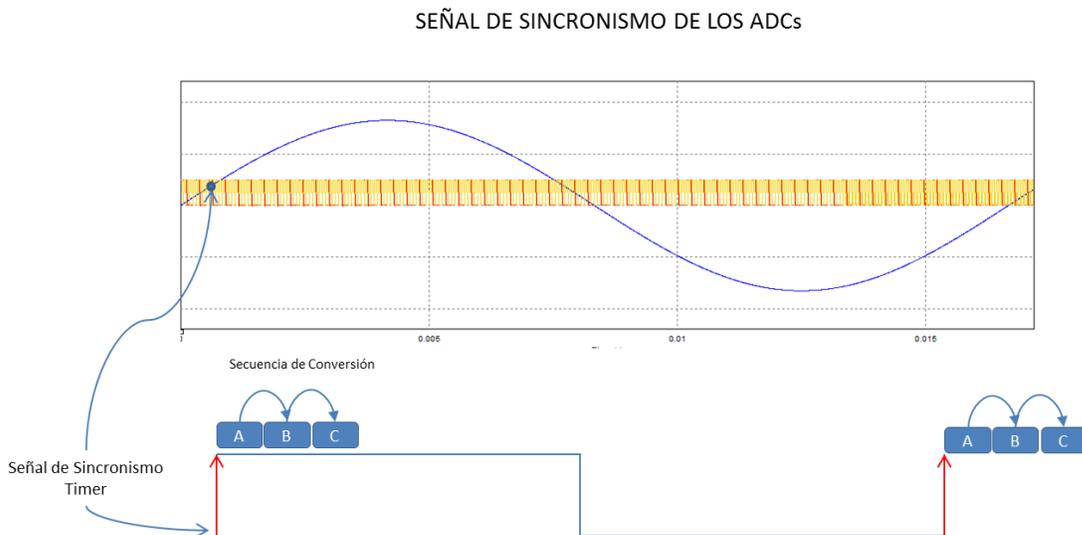


Figura 5.12. Señales de sincronismo de cada ADC, producidas por el Timer

Esta señal es generada por un Timer, programado en UP (conteo ascendente), con Reset en el primer comparador y Set en el segundo para producir la onda cuadrada de frecuencia 3840 Hz. Los dos Timers tienen una programación idéntica.

5.2.1.2 Módulo de Lectura datos V,I. Etapa 3

Corresponde a este módulo, tomar las muestras de voltaje y corriente de cada una de las fases y almacenarlas en buffers de memoria o arreglos. Las tareas implementadas realizan una copia de cada buffer, para trabajar con ella mientras el buffer se está llenando con nuevos datos.

La figura 5.13 ilustra, el funcionamiento, de llenado de los buffer de datos. En este caso se da el ejemplo en el cual el tamaño del buffer es de 64 muestras por señal, pero debemos recordar que puede ser tan pequeño como 4 muestras consiste en crear arreglos de voltaje y corriente e ir llenándolos con las muestras tomadas por el ADC y entregadas en memoria por el DMA. Una vez se da inicio a los Timers, estos generan la señal de trigger al ADC y este comienza la adquisición y conversión de cada dato. Cuando el ADC termina de convertir un dato, su respectivo canal de DMA, se encarga de leer el dato almacenarlo en memoria como lo ilustra la figura 5.13

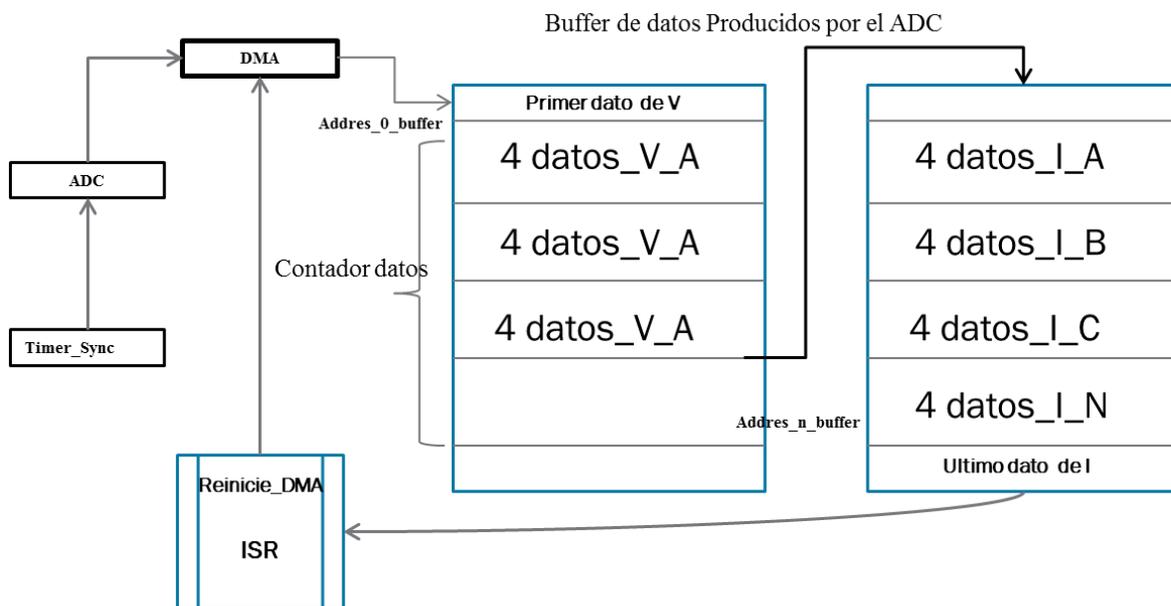


Figura 5.13. Generación de datos desde el ADC hacia la memoria

La rutina de interrupción, debe reiniciar los valores de dirección y tamaño del Buffer, para que el Buffer quede listo para el siguiente ciclo de muestras. El Timer de sincronismo, nunca para.

También este módulo se encarga de convertir los datos muestreados en valores de voltaje y corriente aplicando las respectivas constante de transformación a cada señal y de acumular estos valores de voltaje y corriente en arreglos, durante un segundo, para pasarlos al módulo de cálculos cuando haya transcurrido ese periodo de tiempo.

Cuando se termina de muestrear un Buffer, es decir 4 x7 muestras por este caso (El Buffer ser de mayor tamaño si se desea y depende de los recursos de memoria), la CPU recibe una señal de llenado del Buffer, para que obtenga una copia de Backup, con la cual debe hacer los cálculos.

El proceso de Background, entonces, separa los datos de cada fase, los convierte en voltaje aplicando las constantes de conversión y comienza a acumular la sumatoria de los cuadrados de voltaje y corriente de cada fase y el producto de voltaje por corriente de cada fase.

Cuando este proceso se ha repetido, el número de veces correspondientes a 1 segundo (Si el buffer es de 4 muestras, serán 960 veces), el proceso de Background, genera una interrupción y mediante ella avisa al proceso de Foreground, que los valores de $\sum V^2$, $\sum I^2$, $\sum VxI$, están listos para ser procesados.

El módulo, a su vez va guardando en diferentes variables, los valores acumulados de $\sum V^2$, $\sum I^2$, $\sum VxI$, para pasarlos al proceso de foreground cuando se complete 1 segundo de toma de muestras

En este proceso, la intervención de la CPU es mínima y se limita a realizar la copia de los datos de cada buffer y a realizar los cálculos de las sumatorias antes mencionadas. El trabajo de llenado de los Buffers, lo realiza el DMA, que en el caso del SAM4E16E, se denomina PDC (Peripheral DMA Controller) .

La figura 5.14, ilustra el funcionamiento del módulo de Background implementado.

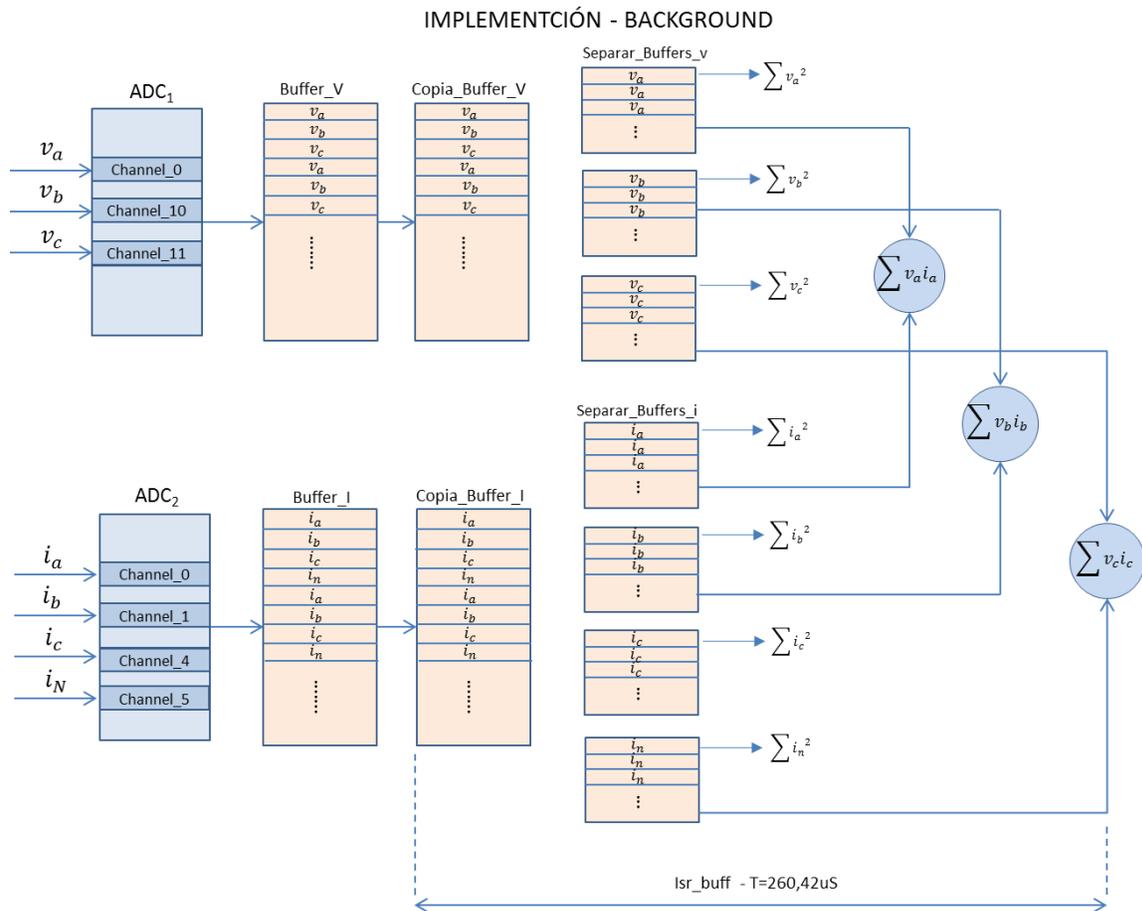


Figura 5.14. Implementación del proceso de background

5.2.2 Proceso de Foreground. Etapa 4

Corresponde a este proceso, realizar los cálculos de parámetros de acuerdo a los datos recibidos del proceso de background, cuando este ha completado el tiempo de un segundo. Adicionalmente este es el proceso general que se encarga de inicializar el sistema, es decir de configurar su reloj, configurar todos los periféricos como ADC, DMA (PDC), TIMER, RTC, UART, GPIO, LCD, SD, etc. Adicionalmente se encarga de las interfaces de comunicación y de usuario, es decir de transmitir los datos calculados hacia el display LCD, hacia la terminal serial y hacia la terminal remota vía ZigBee. El proceso de foreground, está compuesto por los módulo de cálculos, control de consumo, comunicaciones, interface de usuario. El diagrama general de flujo del proceso se ilustra en la figura 5.15

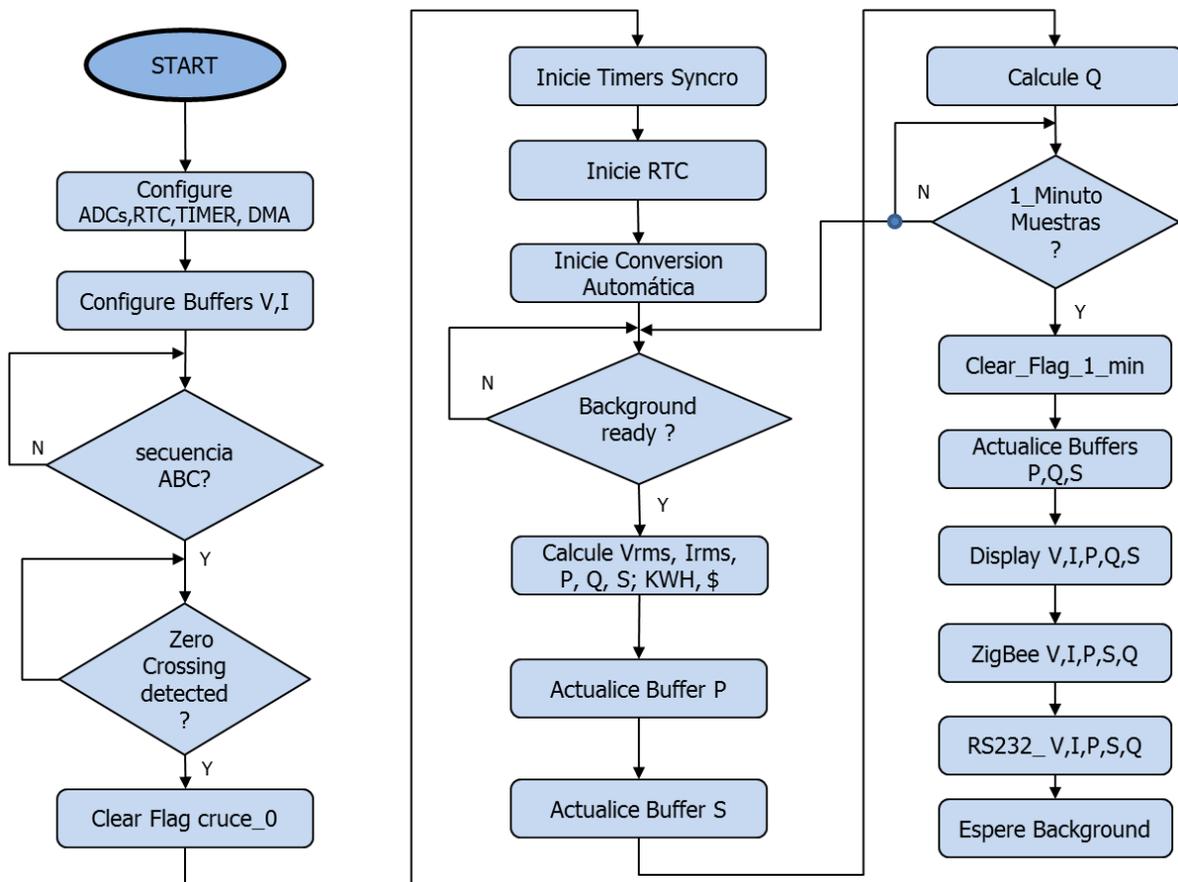


Figura 5.15. Diagrama de Flujo del Proceso de Foreground

La implementación de cada uno de las tareas ilustradas en el diagrama de flujo de la figura 5.15, se muestra en la figura 5.16

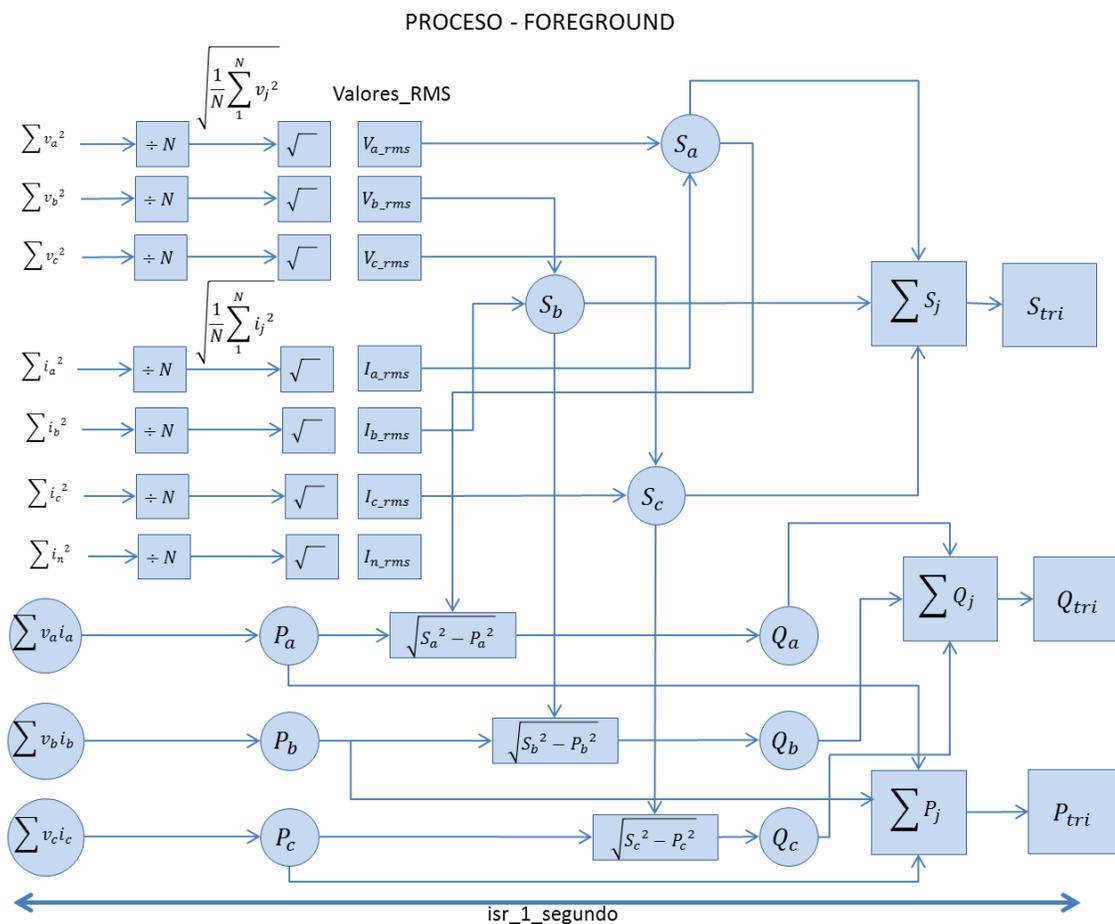


Figura 5.16. Implementación del Proceso de foreground

5.2.2.1 Módulo de Cálculos

Este módulo es el encargado de aplicar las ecuaciones correspondientes, para realizar los cálculos de voltaje y corriente rms de cada una de las fases, de potencia activa en cada fase, potencia aparente en cada fase, potencia reactiva por fase, potencia total acumulada, energía acumulada y valor del consumo en pesos. Las ecuaciones aplicadas para realizar estos cálculos se ilustran en la tabla 5.1.

CALCULOS DE POTENCIA Y ENERGIA

SEÑAL CONTINUA	SEÑAL DISCRETA
VOLTAJE o CORRIENTE RMS $V_{RMS} = \sqrt{\frac{1}{T} \int_0^T (v(t))^2 dt}$	VOLTAJE o CORRIENTE RMS $V_{rms} = \sqrt{\frac{1}{N} \sum_{j=1}^N v_j^2} \quad (C1)$
POTENCIA ACTIVA $P = \frac{1}{T} \int_0^T v(t) i(t) dt$	POTENCIA ACTIVA $P_{ph} = \frac{1}{N} \sum_{n=1}^N v_{ph}(n) i_{ph}(n) \quad (C2)$ <p style="text-align: right; font-size: small;"><i>ph</i> → fase A, fase B, fase C</p>
POTENCIA REACTIVA $Q = \frac{1}{T} \int_0^T v(t) i(t) \sin(\theta) dt$	POTENCIA REACTIVA $Q = \sqrt{S^2 - P^2} \quad (C3)$
POTENCIA APARENTE $S = V_{RMS} I_{RMS}$ $S = \sqrt{P^2 + Q^2}$	POTENCIA APARENTE $S = V_{RMS} I_{RMS} \quad (C4)$ $S = \sqrt{P^2 + Q^2}$
ENERGIA $dE = P dt$ $E_t = Pxt$	ENERGIA $E_{act} = \sum_{n=1}^N v_{ph}(n) i_{ph}(n) \quad (C5)$

Tabla 5.1. Fórmulas aplicadas para calcular Voltaje, Corriente y Potencia con señales discretas

Para el calculo de valores RMS, se aplica la ecuación (C1), tomando los valores de $\sum V^2, \sum I^2$ entregados por el proceso de background. El Valor rms de voltaje y corriente es utilizado para calcular el valor de potencia aparente cada segundo, mediante la ecuación (c4a). Ese cálculo se hace para cada fase. La ecuación del voltaje rms correspondiente al número de muestras tomadas es

$$V_{rms_k} = \frac{1}{3840} \sum_{n=1}^{3840} v_{ph_k}(n)^2 \quad (5.16)$$

La ecuación para el cálculo de corriente rms es similar a la ecuación (5.16).

$$I_{rms_k} = \frac{1}{3840} \sum_{n=1}^{3840} I_{ph_k}(n)^2 \quad (5.17)$$

El valor de potencia por fase es guardado y acumulado, para que esté disponible para consulta. Lo mismo se hace para la potencia total.

La potencia activa se calcula, mediante la fórmula (C2). esta potencia se va acumulando en una variable, para que siempre se tenga tanto el último valor como el consumo de potencia acumulada.

La potencia activa total, se calcula sumando las potencias activas de las tres fases, es decir mediante la ecuación

$$P_T = \sum_1^3 \frac{1}{N} \sum_{n=1}^{3840} v_{ph}(n) i_{ph}(n) = P_A + P_B + P_C \quad (5.18)$$

La potencia reactiva se puede calcular de dos formas: Mediante la ecuación (C3), o desfasando el voltaje en 90° ($\pi/4$) y realizando una sumatoria similar a la de la ecuación (C2).

La potencia aparente total se calcula a partir de la potencia aparente de cada fase

$$S_T = \sum_{j=1}^3 V_{j_rms} I_{j_rms} = S_A + P_B + P_C \quad (5.19)$$

La Potencia aparente se calcula desfasando el voltaje 90° o con el triangulo de potencia

$$Q_T = \sum_1^3 \frac{1}{N} \sum_{n=1}^{3840} v_{ph} 90(n) i_{ph}(n) = P_A + P_B + P_C \quad (5.20)$$

La energía, se calcula, de tal forma que quede expresada en KW-H, como establece la norma IEC62053-21. Es decir, se toma la potencia activa total (En un Smart Meter industrial, se toma la potencia aparente total), de cada segundo y se multiplica por el número de muestras tomadas durante el segundo. En nuestro caso 3840. Ese valor se va acumulando, hasta completar el acumulado de un minuto. Cuando el RTC, genera la interrupción de un minuto, el proceso de foreground, envía hacia el LCD , la terminal y vía ZigBee, el valor de Energía total acumulada.

El proceso, continúa acumulando valores de potencia y energía de forma indefinida. No hay problemas de memoria, porque los valores se guardan en una dos variables, una que contiene el último valor y otra que contiene el valor acumulado.

5.2.2.2 Módulo de Control

Este módulo, tiene el único propósito, de comparar, que los consumos de energía y valor del consumo, no sobrepasen valores previamente establecidos.

Se implementan funciones de comparación, con valores de referencia de Potencia activa total acumulada, Potencia total acumulada, energía total acumulada y valor de consumo total acumulado, para periodos de 24 horas mínimo. Se pueden programar periodos más cortos, si se desea.

5.2.3 Módulo de Interface a Usuario. Etapa 5

Para mostrar los datos en display, se utiliza el LCD que viene en la tarjeta de desarrollo SAM4E-EK. Los datos básicos que se muestran son los valores de voltaje, Corriente, Potencia activa total, Energía total acumulada del último minuto y el valor en pesos del consumo total acumulado.

5.2.4 Módulo de Comunicaciones ZigBee y serial. Etapa 5

Contiene todas las tareas necesarias para transmitir de forma inalámbrica los datos acumulados, hacia una terminal serial con interface RS232/485, para facilitar la visualización de los mismos.

El diagrama de flujo de la implementación del módulo ZigBee, se ilustra en la figura 5.16

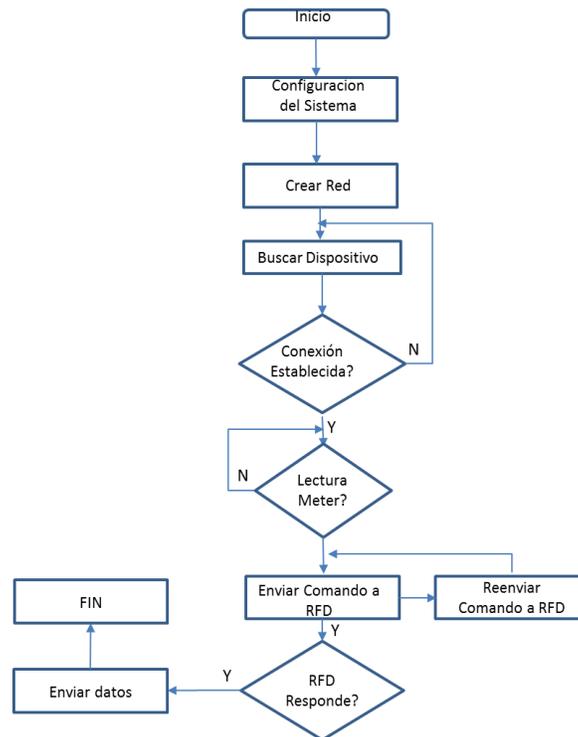


Figura 5.16 Diagrama de flujo de la comunicación entre los dos nodos.

CONFIGURACION DE DISPOSITIVOS XBee

Se realizará utilizando el software XCTU de Digi International

XCTU es un software que permite programar cada dispositivo, XBee mediante una interfaz gráfica: Se puede configurar y probar cada módulo, antes de ponerlo en funcionamiento. Los módulos Xbee, se pueden programar de dos formas: Modo Transparente y Modo API.

En el modo transparente, el Xbee se comporta como una línea serial que recibe y envía los datos, por una UART. En modo API, se implementa la transmisión de datos por tramas, donde se definen las operaciones y eventos que se quieren transmitir. Como en nuestro caso únicamente se necesitan 2 nodos, su implementación se hace en modo transparente.

La configuración, se hace creando una red con una dirección común a los dos radios. El primer radio (Aquel, que se encuentra en la tarjeta AFE), conectado a la tarjeta de desarrollo, vía UART es el ZigBee COORDINATOR y el segundo radio, conectado remotamente a una terminal por ejemplo, es el END DEVICE. La figura 5.17 ilustra la configuración de los radios

The image shows two screenshots of the XCTU configuration interface for ZigBee devices. Each device configuration is presented in a blue header box with a 'RF ZB' logo and a list of parameters in a table below.

Device 1: ENDDEVICE1 (ZigBee End Device AT)

Name:	ENDDEVICE1
Function:	ZigBee End Device AT
Port:	COM3 - 9600/8/N/1/N - AT
MAC:	0013A20040B451C0

ID PAN ID	1234	SH Serial Number High	13A200
SC Scan Channels	FFFF Bitfield	SL Serial Number Low	40B451C0
SD Scan Duration	3 exponent	MY 16-bit Network Address	FFFE
ZS ZigBee Stack Profile	0	MP 16-bit Parent Address	FFFE
NJ Rejoin Policy	FF	DH Destination Address High	13A200
JN Join Notification	Disabled [0]	DL Destination Address Low	40B45109
OP Operating PAN ID	0	NI Node Identifier	ENDDEVICE1
OI Operating 16-bit PAN ID	FFFF		
CH Operating Channel	0		

Device 2: ROOT (ZigBee Coordinator AT)

Name:	ROOT
Function:	ZigBee Coordinator AT
Port:	COM5 - 9600/8/N/2/N - AT
MAC:	0013A20040B45109

ID PAN ID	1234	SH Serial Number High	13A200
SC Scan Channels	FFFF Bitfield	SL Serial Number Low	40B45109
SD Scan Duration	3 exponent	MY 16-bit Network Address	0
ZS ZigBee Stack Profile	0	DH Destination Address High	13A200
NJ Node Join Time	FF x 1 sec	DL Destination Address Low	40B451C0
OP Operating PAN ID	1234	NI Node Identifier	ROOT
OI Operating 16-bit PAN ID	7AC0		
CH Operating Channel	18		
NC Number of Re...ing Children	9		

Figura 5.17 Configuración de los radios ZigBee

Se ha configurado una red, con dirección 1234. Ambos radios deben tener la misma ID de red. La dirección del radio COORDINADOR o ROOT es la 40B45109 y la del END DEVICE 40B451C0. La parte baja de la dirección de destino de cada radio debe ser igual a la parte baja de la dirección source del otro, como muestra la figura 5.17.

PAYLOAD: El tamaño máximo recomendado de cada paquete de transmisión es de 100 Bytes , se debe respetar este límite para evitar la pérdida de datos, ya que el límite está determinado por el Buffer de entrada de cada radio.

Los datos se transmiten a través de las interfaces implementadas en módulo UART, del programa, el cual corresponde a la implementación del código para transmisión vía UART.

Los datos transmitidos hacia la terminal, llegan sin ninguna distorsión y en los tiempos deseados, lo cual confirma la adecuada selección de este estándar, para transmitir datos de energía.

6 ANALISIS DE RESULTADOS

6.1 Calibración de ADCs

Se realizó, para cada canal de cada ADC que se utilizó. La metodología consistió en suministrar tres niveles de voltaje DC a cada canal, mediante una fuente y ver los resultados de la conversión del valor muestreado. Se tomaron valores cercanos a los 27 mV por ser el valor mínimo a medir, valores medios de aproximadamente 1.5 V y el valor máximo del ADC de 3.3 V.

Las figura 6.1a, 6.1b y 6.1c ilustran las imágenes de las mediciones realizadas por el ADC, comparadas por las de un multímetro de precisión marca KEITHLEY modelo 2110.

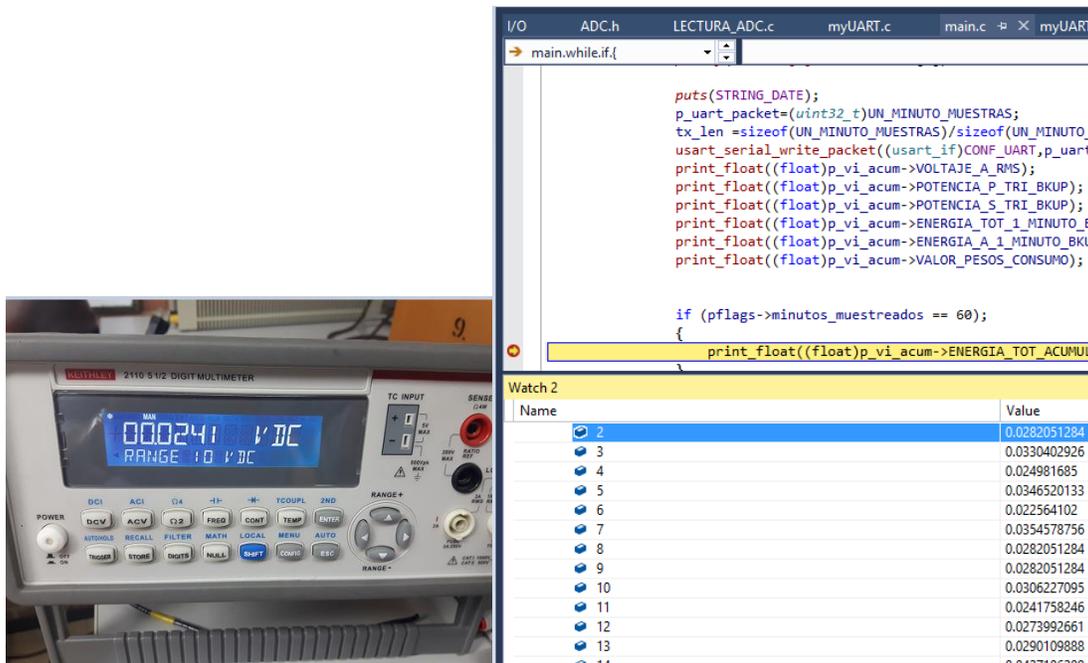


Figura 6.1a. Calibración ADC1, valores pequeños

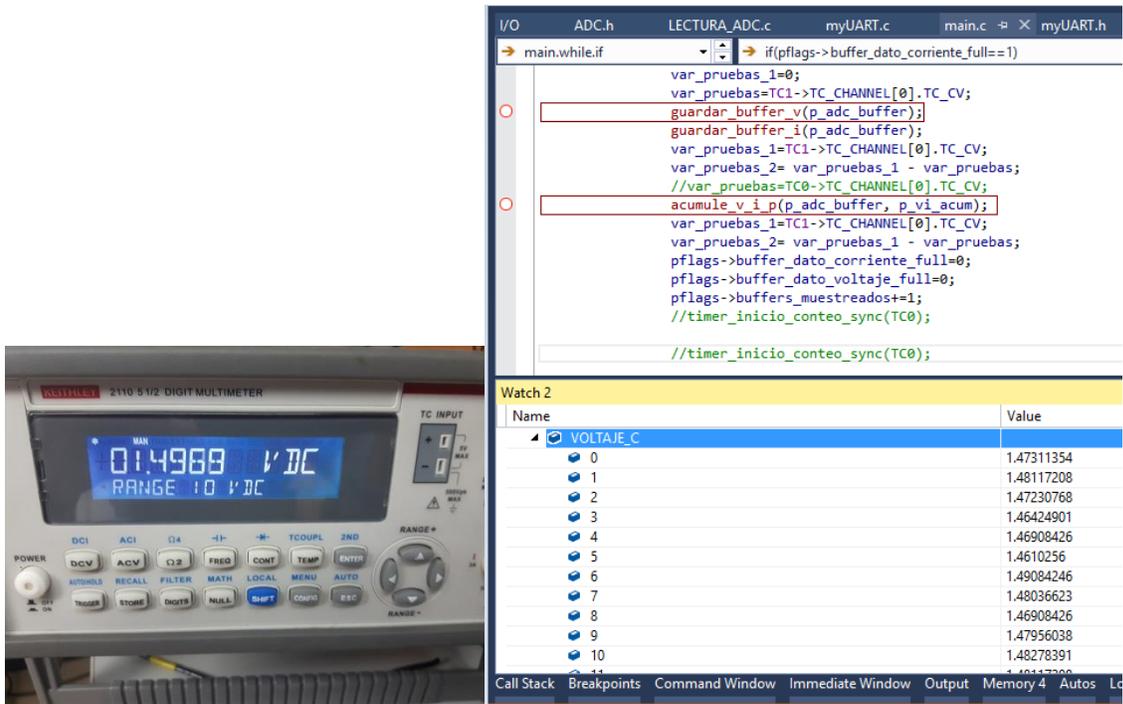


Figura 6.1b. Calibración ADC1, valores medios

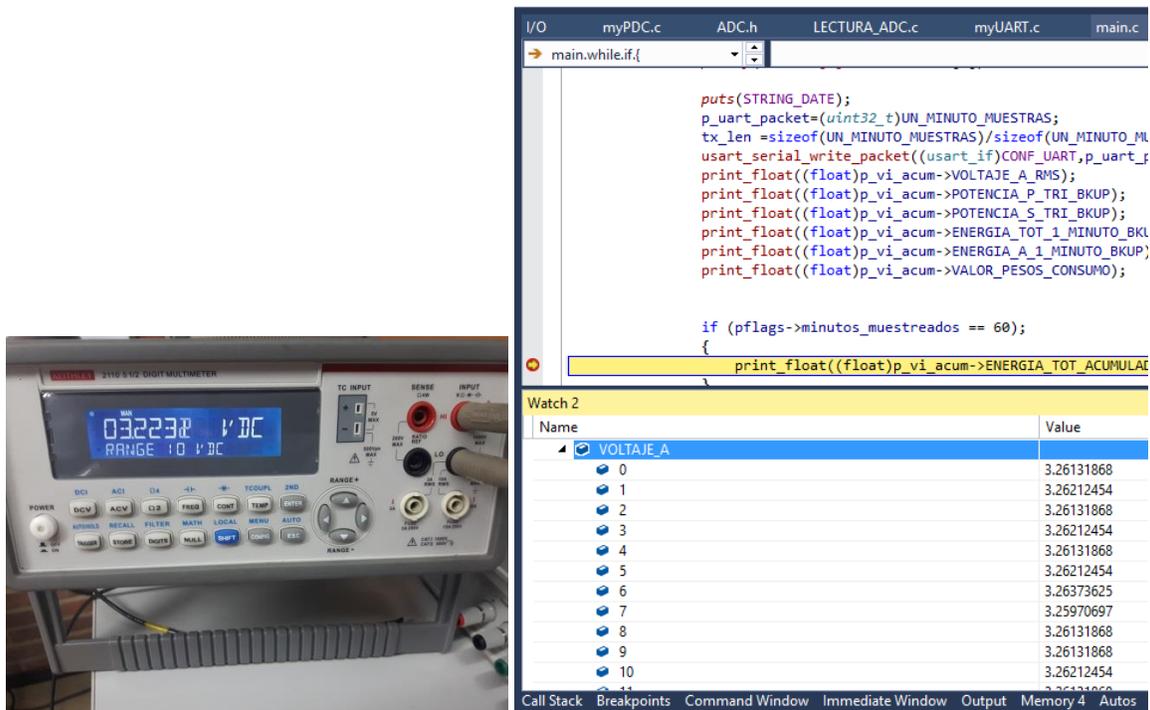


Figura 6.1c. Calibración ADC1, valores máximos

Las mayores desviaciones en las mediciones se presentan en el valor mínimo del ADC. Esto era de esperarse, porque es en esos valores donde la señal tiene más ruido.

Este proceso se realizó para todos los canales y los resultados se muestran en la tabla 6.1

ADC 1								
MULTIMETRO	CH_0	ERROR %	CH_1	ERROR %	CH_4	ERROR %	CH_5	ERROR %
0,0241	0,0260	7,31	0,0258	6,59	0,0257	6,23	0,0259	6,95
1,4969	1,4722	-1,68	1,4783	-1,26	1,4765	-1,38	1,4789	-1,22
3,2231	3,2621	1,20	3,2628	1,22	3,2609	1,16	3,2642	1,26
ADC 2								
MULTIMETRO	CH_0	ERROR %	CH_1	ERROR %	CH_2	ERROR %	CH_3	ERROR %
0,0241	0,0256	5,86	0,0255	5,49	0,0259	6,95	0,0255	5,49
1,4969	1,4735	-1,59	1,4792	-1,20	1,4708	-1,77	1,4718	-1,71
3,2231	3,2604	1,14	3,2645	1,27	3,2687	1,40	3,2674	1,36

Tabla 6.1 . Calibración de los ADCs. Medidas en Voltios

6.2 Adecuación de señales:

Las figura 6.2 ilustra el primer paso de acondicionamiento de la señal AC de entrada de voltaje de la fase A.

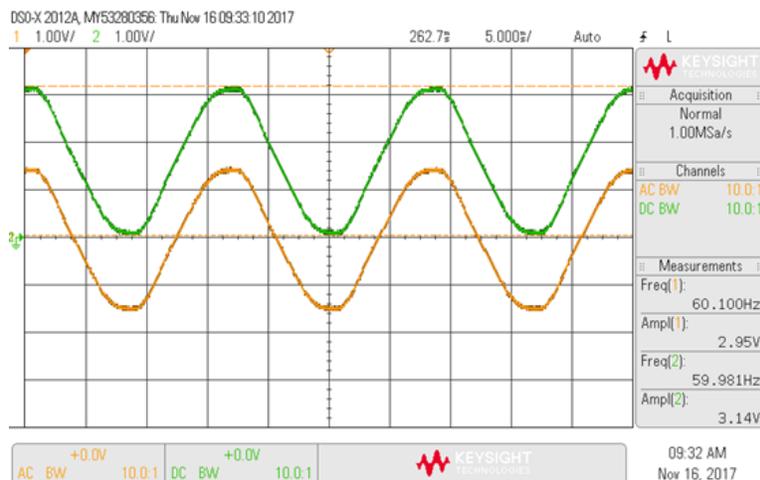


Figura 6.2 Señal de entrada con level shifter

Primero la señal se reduce por el transformador de entrada. Puede verse la señal original (señal AC) y la señal sin nivel AC, una vez se pasa por el level shifter, para que su valor quede comprendido entre 0 y 3.3 V, que es el máximo valor permitido por el ADC.

La figura 6.3 ilustra la señal, de entrada al ADC, una vez ha sido pasada por el filtro RC, puede observarse el desfase que produce el filtro. Este desfase debe ser compensado, digitalmente.

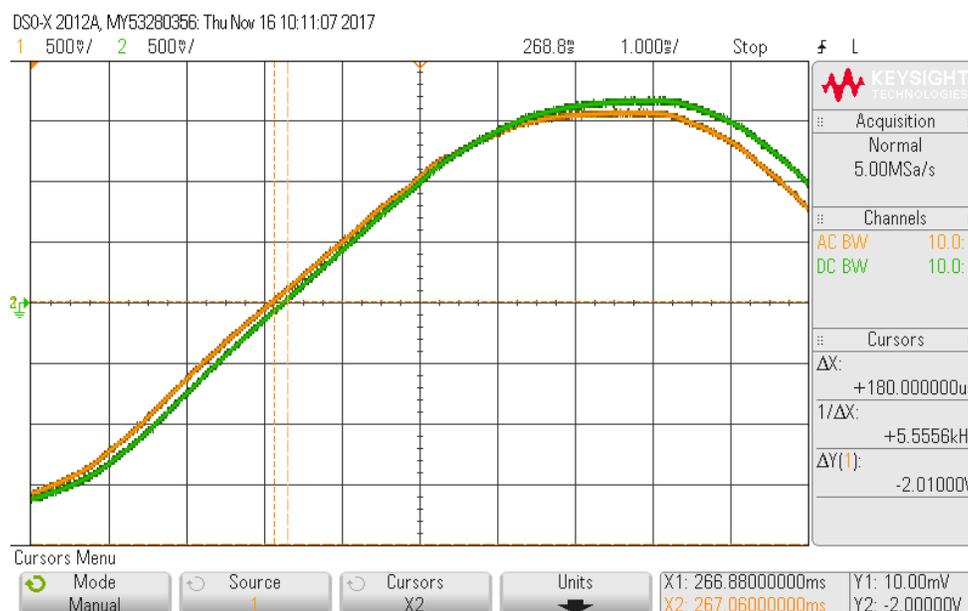


Figura 6.3 Desfase producido por el filtro anti-aliasing.

La figura 6.4 muestra la señal de sincronismo, para el inicio de conversiones.

6.3 Método sub_buffering:

La reducción de espacio en memoria, esperada con el método sub-buffering, es muy buena. Las imágenes (screen-shots), del programa principal una vez compilado, muestran los distintos valores que ocupa la memoria, para diferentes tamaños de memoria RAM. Las figuras 6.4 , 6.5 y 6.6, ilustran algunos valores de prueba de tamaño de l buffer Y la tabla 6.3, muestra los posibles valores para un máximo de 256 muestras por ciclo, con un buffer de 256 muestras y dos ciclos de acumulación de datos. Debe aclararse, que el sistema no funciona en esta última opción.

```

ADC.h  x  LECTURA_ADC.c  myUART.c  main.c  myUART.h  conf_uart_serial.h
-----*/
#define PI_VALUE_MIO      3.141592
#define RAIZ_DE_DOS      1.414214
#define k_adc             (float32_t)VREF/MAX_DIGITAL_VALUE

#define   FREQ_RED        60
#define   NUM_MUESTRAS_CICLO  4           // Este dato depende de la RAM del
// Con 10 ciclos se tendran 2x7x10x
//Multiplo de 2 a la n para facilit
//EL Buffer debe llenarse cada 1 ci

#define   NUM_CICLOS_BUFFER  (float32_t)1

//#define   FREQ_MUESTREO    (uint32_t)FREQ_RED*NUM_MUESTRAS_CIC
#define   NUMERO_CICLOS     2           //Numero de ciclos a muestrear para
-----*/

100 %
Output
Show output from: Build
Done executing task "RunCompilerTask".
Task "RunOutputFileVerifyTask"
  Program Memory Usage : 42608 bytes  4,1 % Full
  Data Memory Usage   : 17016 bytes  13,0 % Full
Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "METER_MONO_ATMEL_1.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') wa
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets"
Done building target "Build" in project "METER_MONO_ATMEL_1.cproj".
Done building project "METER_MONO_ATMEL_1.cproj".

Build succeeded.
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
Output  Error List  Breakpoints

```

Figura 6.4. Método sub-buffering con 4 muestras , 2 ciclos

```

ADC.h  x  LECTURA_ADC.c  myUART.c  main.c  myUART.h  conf_uart_serial.h
-----*/
#define PI_VALUE_MIO      3.141592
#define RAIZ_DE_DOS      1.414214
#define k_adc             (float32_t)VREF/MAX_DIGITAL_VALUE

#define   FREQ_RED        60
#define   NUM_MUESTRAS_CICLO  32        // Este dato depende de la RAM del
// Con 10 ciclos se tendran 2x7x10x
//Multiplo de 2 a la n para facilit
//EL Buffer debe llenarse cada 1 ci

#define   NUM_CICLOS_BUFFER  (float32_t)1

//#define   FREQ_MUESTREO    (uint32_t)FREQ_RED*NUM_MUESTRAS_CIC
#define   NUMERO_CICLOS     2           //Numero de ciclos a muestrear par
-----*/

100 %
Output
Show output from: Build
Done executing task "RunCompilerTask".
Task "RunOutputFileVerifyTask"
  Program Memory Usage : 42608 bytes  4,1 % Full
  Data Memory Usage   : 19144 bytes  14,6 % Full
Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "METER_MONO_ATMEL_1.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') wa
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets"
Done building target "Build" in project "METER_MONO_ATMEL_1.cproj".
Done building project "METER_MONO_ATMEL_1.cproj".

Build succeeded.
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
Output  Error List  Breakpoints

```

Figura 6.4. Método sub-buffering con 32muestras , 2 ciclos

```

ADC.h  LECTURA_ADC.c  myUART.c  main.c  myUART.h  conf_uart_serial.h
...
#define PI_VALUE_MIO 3.141592
#define RAIZ_DE_DOS 1.414214
#define k_adc (float32_t)VREF/MAX_DIGITAL_VALUE

#define FREQ_RED 60
#define NUM_MUESTRAS_CICLO 256 // Este dato depende de la RAM de
// Con 10 ciclos se tendran 2x7x1
// Multiplo de 2 a la n para facil
#define NUM_CICLOS_BUFFER (float32_t)1 //EL Buffer debe llenarse cada 1

// #define FREQ_MUESTREO (uint32_t)FREQ_RED*NUM_MUESTRAS_C
#define NUMERO_CICLOS 2 //Numero de ciclos a muestrear pa
...
Output
Show output from: Build
Done executing task RunCompilerTask .
Task "RunOutputFileVerifyTask"
Program Memory Usage : 42672 bytes 4,1 % Full
Data Memory Usage : 36168 bytes 27,6 % Full
Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "METER_MONO_ATMEL_1.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '')
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets"
Done building target "Build" in project "METER_MONO_ATMEL_1.cproj".
Done building project "METER_MONO_ATMEL_1.cproj".

Build succeeded.
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====

```

Figura 6.4. Método sub-buffering con 256 muestras , 2 ciclos

Las imágenes pueden observarse en su tamaño original, en los anexos del documento.

6.4 Mediciones de Voltaje, Corriente y Potencia.

Las primeras pruebas se realizaron con una carga monofásica, con voltaje variable (Mediante un variac monofásico) entre 102 y 135 VAC. Los resultados de las mediciones de voltaje realizadas con el multímetro KEITHLEY 2110 y las de corriente efectuadas con la pinza AGILENT N2783B, se ilustran en la tabla 6.2

ERROR EN PORCENTAJE PARA VOLTAJE Y CORRIENTE VARIABLE EN UNA FASE								
Vin (V)	Vadc	ERROR V	I	Iadc	ERROR I	P	Padc	ERROR P
102,3560	99,8990	2,4004	0,3412	0,3317	2,7843	34,9239	33,1365	5,1179
105,491	103,5920	1,8002	1,0800	1,0500	2,7778	113,9303	108,7716	4,5279
108,233	106,7180	1,3998	2,1200	2,0600	2,8302	229,4540	219,8391	4,1903
112,3140	111,4150	0,8004	3,0800	3,0345	1,4773	345,9271	338,0888	2,2659
115,682	116,8150	-0,9794	4,1400	4,0300	2,6570	478,9235	470,7645	1,7036
118,243	119,8030	-1,3193	5,0900	5,0100	1,5717	601,8569	600,2130	0,2731
121,972	119,4100	2,1005	6,0500	6,1200	-1,1570	737,9306	730,7892	0,9678
125,684	126,7520	-0,8498	7,0800	7,1500	-0,9887	889,8427	906,2768	-1,8469
130,215	127,1650	2,3423	8,0600	7,9470	1,4020	1049,5329	1010,5803	3,7114

Tabla 6.2. Mediciones monofásicas con sus respectivos errores

Posteriormente se efectuaron pruebas con una carga trifásica, con voltaje variable en una de sus fases y fijo en las otras dos. Se instaló un VARIAC monofásico, conectado a la fase A, en cuya salida se instaló un reóstato de carga con valor en 8Ω , se hicieron incrementos aproximados de 5 voltios a la salida del VARIAC para obtener variaciones en corriente. El valor de resistencia de carga para la fase B fue de 10Ω y el de la fase C de 7.5Ω .

Se puso el sistema en operación y se tomaron datos durante 10 minutos, para obtener valores apreciables de Potencia y energía.

Se tomaron las medidas de Voltaje con el Multímetro de precisión KEITHLEY 2110 y de Corriente con la pinza AGILENT N2783B. La tabla 6.3 muestra los resultados obtenidos.

PARAMETRO	MULTIMETRO	SISTEMA	ERROR %
Va	124,13	126,21	-1,68
Vb	123,56	125,23	-1,35
Vc	122,85	124,94	-1,70
Ia	13,18	12,92	1,97
Ib	12,33	11,96	3,00
Ic	16,56	16,21	2,11
Ptot	5,82	5,73	1,55
Qtot	5,82	5,73	1,55
Stot	5,82	5,73	1,55
KWH	0,97	0,96	1,44
Costo	388,00	384,00	1,03

Tabla 6.3 Valores de Potencia (VA) y Consumo para 10 minutos

6.5 Transmisión de datos e interface de usuario:

Los datos , de Potencia, Energía y consumo, son transmitidos por el módulo ZigBee y se muestran en una terminal con el software de recepción XCTU suministrado por Digi International. La figura 6.5 ilustra un envío de datos medidos cada minuto.

Puede observarse la correcta transmisión de los datos, lo cual ratifica la confiabilidad de este tipo de dispositivo, en estas aplicaciones.

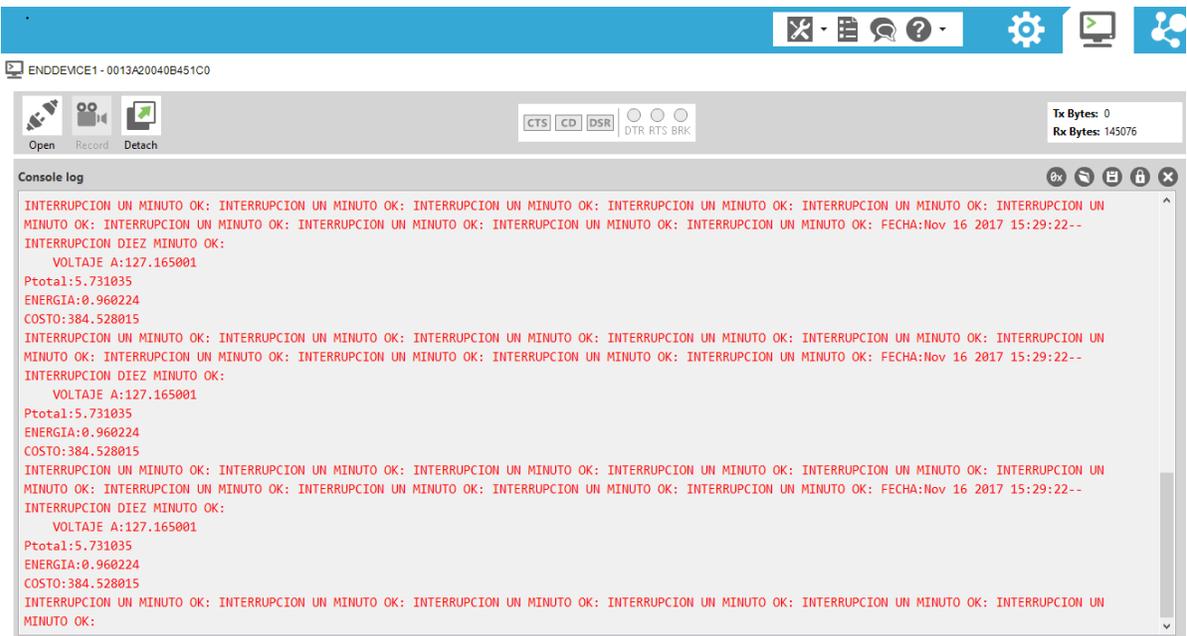


Figura 6.5. Datos Enviados Vía ZigBee cada 10 minutos

7 CONCLUSIONES

De los resultados obtenidos, podemos obtener las siguientes conclusiones:

- En cuanto a la precisión de las medidas el sistema propuesto, cumple con las especificaciones establecidas por el estándar tomado como referencia: IEC 62053-21 Clase 2, que permite variaciones hasta del 2.5% en Voltaje y Corriente.
- La precisión se puede mejorar, teniendo en cuenta que los ADC, se trabajaron con su resolución original que es de 12 bits, pero esta se puede mejorar, realizando oversampling, para llevarla a 16 bits. Esto únicamente incrementaría las tareas de software, porque sería necesario un proceso de DECIMACIÓN de la señal digital, para volverla a su número de muestras original.
- Se puede mejorar la adecuación de señales, incluyendo un filtro activo. Esto contribuye a tener una mejor lectura de voltajes bajos a la entrada del ADC.
- Se observa, como con 2 ADCs es posible muestrear una señal trifásica. Esto permite seleccionar un gran número de microcontroladores para realizar medidores inteligentes acordes con una necesidad específica.
- En este proyecto, no se pudieron utilizar los ADC en modo diferencial, por falta de disponibilidad de los pines en la tarjeta de desarrollo. Esto también mejoraría la calidad de las mediciones, porque el modo diferencial, filtra el ruido de modo común.
- La apropiada cantidad de periféricos y sus características, son fundamentales a la hora de implementar un Smart Meter. Este es un criterio que se debe aplicar de forma rigurosa cuando se realice la selección.
- Es muy importante la selección de los sensores de medición de las señales. Por ejemplo, para mediciones de voltaje, es preferible optar por resistencias shunt, que evitan los desfases y distorsiones de la señal existentes en los transformadores de voltaje. En este proyecto se optó por tener transformadores de aislamiento para proteger la tarjeta de desarrollo que tiene un costo alto
- La medición de armónicos queda como un futuro trabajo para incluir en el sistema. esto lo convierte en un Smart Meter con una mayor gama de aplicaciones. El software actual, permite medir hasta el 32nd, armónico
- El método sub-buffering, es un logro muy importante. Se reduce considerablemente la utilización del recurso de memoria en el sistema. Incluso se realizaron pruebas, con todo el programa y los datos corriendo en memoria RAM y funciona perfectamente. Este es un MCU de 128 KB de RAM
- Se ha diseñado e implementado un sistema, que para su futura construcción, exige unas especificaciones técnicas reducidas y de fácil obtención en la actualidad: CPU Cortex M4 de 16MHz mínimo, pocos periféricos: 2 ADC, un DMA de 2 canales, un RTC, 3 TIMERS, 64K B Flash,

8 BIBLIOGRAFIA Y FUENTES DE INFORMACION

- [1] The Smart Grid An Introduction, prepared for the U.S. Department of Energy by Litos Strategic Communication www.litoss.com/energy
- [2] J.C.P. Kester, Maria José González, John Parsons, Smart Metering Guide, Energy Saving and the Customer, Edition 2010, Energy research Center of Netherlands ,ECN
- [3] European Smart Metering Alliance Final Report, ESMA, BEAMA and others.
- [4] Smart Meters and Smart Meter Systems: A Metering Industry Perspective, Edison Electric <http://smartgrid.eei.org/Pages/resources.aspx>
- [5] Smart Metering and Home Automation Solutions for the Next Decade, Shafik Ahmad, IEEE.
- [6] MSP430F6736 Single-phase energy meter IC System on Chip, Texas instruments
- [7] MSP430F677x Ultra-Low Power Polyphase Energy Meter System on Chip, Texas Instruments
- [8] Evaluating the ADE7878 Energy Metering IC, Analog Devices
- [9] China's Electricity Meter Market to Reach 110.6 Million Units in 2012 <http://www.isuppli.com/China-Electronics-Supply-Chain/MarketWatch/Pages/China%E2%80%99s-Electricity-Meter-Market-to-Reach-110.6-Million-Units-in-2012.aspx>
- [10] Consumer and Energy Efficiency Benefits of Smart Metering, ESMIG, European Smart Metering Industry Group
- [11] Home networks and residential gateways – World 2012 www.ihs.com
- [12] Shahin Farahani, Zigbee Wireless Networks and transceivers, 2008 Elsevier
- [13] Meera Balakrishnan, Smart Energy Solutions for Home Area Networks and Grid-End Applications, Freescale www.freescale.com
- [14]. Smart Metering – isuppli : <http://www.isuppli.com/Pages/Market-Research-Search.aspx?k=smart%20metering>. A Octubre 2017
- [15] Bart Basile, Stefan Schauer, Kripasagar Venkat, Implementation of a Single-Phase Electronic Watt-Hour Meter Using the MSP430F6736.
- [16] MQX-Enabled MK30X256 Single Phase Electricity Meter reference Design, FREESCALE
- [17] MCP 3909/dsPIC33FJ128GP206 3-Phase Energy Meter Reference Design, MICROCHIP TECHNOLOGY
- [18] Underwood, Franglin, Chan, Three Phase Electronic Watt-Hour Meter Design Using MSP430.
- [19] Wang Jiahui, Liu Xiaodang, Zeng Lei, Hou Weiyan, The Design an Implementation of a Wireless Meter reading System, ICEMI 2011.
- [20] International Standard IEC-62053 -21 Static meters for active energy, first edition 2003-01
- [21] ZigBee Alliance, www.zigbee.org
- [22] Design an Implementation of Smart Three Phase Energy Meter
- [23] Nokhil Patil, y Dnyaneshwar Bondar, Intelligent Energy Meter with Advanced Billing System and Electricity Theft Detection, IEEE.
- [24] Design of the networked electricity meter based on GPRS
- [25] SAM4E16E Datasheet, Atmel Corporation.
- [26] SAM4E-EK_User-Guide, Atmel Corporation
- [27] ABCs_of_ADCs, Analog-to-Digital Converter Basics
- [29] Figura tomada de Wikipedia: https://en.wikipedia.org/wiki/Rogowski_coil 20/08/2017

[30] Jing Li Yihan Yang Wenbin Yu and Guoqing Zhang, Digital electric energy meter individually measuring electric energy of fundamental wave and harmonic wave based on Lab VIEW DSP module

9 ANEXOS

Se incluyen los siguientes anexos al presente trabajo:

Anexo 1: Copia de los archivos de diseño del PCB en Altium 16

Anexo 2: Imágenes de screen-shots del software y de osciloscopio

Anexo 2: Copia de archivos de software. Consulta.

Estos archivos no pueden ser, copiados, descargados o utilizados por terceros sin la previa autorización de ENERPOWER SAS, quien es la propietaria de los derechos de autor y de explotación del diseño resultado del presente proyecto