

UNIVERSIDAD SAN FRANCISCO DE QUITO

Colegio de Ciencias e Ingeniería

Diseño y desarrollo de una red de sensores inalámbricos para la medición del consumo eléctrico en aplicaciones de bajo voltaje

Santiago Ricardo Mancheno López

Alberto Sánchez, Ph.D., Director de Tesis

Tesis de grado presentada como requisito para la obtención del título de
Ingeniero Electrónico

Quito, julio de 2015

Universidad San Francisco de Quito

Colegio de Ciencias e Ingeniería

HOJA DE APROBACIÓN DE TESIS

Diseño y desarrollo de una red de sensores inalámbricos para la medición del consumo eléctrico en aplicaciones de bajo voltaje

Santiago Ricardo Mancheno López

Alberto Sánchez, Ph.D.,
Director de Tesis

Diego Benítez, Ph.D.,
Miembro del Comité de Tesis

Omar Aguirre, MSc.,
Coordinador del Departamento de Ingeniería
Eléctrica y Electrónica

Ximena Córdova, Ph.D.,
Decana de la Escuela de Ingeniería
Colegio de Ciencias e Ingeniería

Quito, julio de 2015

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído la Política de Propiedad Intelectual de la Universidad San Francisco de Quito y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma: _____

Nombre: Santiago Ricardo Mancheno López

C.I.: 1715246193

Fecha: Quito, julio de 2015

Dedicatoria

Dedico este trabajo a esa persona que fue mi apoyo incondicional en los buenos y malos momentos, de inicio a fin.

Para la compañera de mis noches de insomnio y mis días sin descanso.

Para mi novia Clau.

Agradecimientos

Agradezco a todos los profesores del departamento de ingeniería electrónica de la universidad por toda su paciencia y enseñanza a lo largo de estos años.

Agradezco a todos mis compañeros de carrera con quienes he compartido este largo y duro proceso que me ha permitido realizar satisfactoriamente este trabajo.

Agradezco a mi tutor de tesis por todo su apoyo y confianza desde el inicio hasta el fin de este trabajo.

Agradezco a mi familia por todo su apoyo incondicional el cual fue muchas veces mi sustento en los momentos más difíciles.

Resumen

El presente trabajo de tesis, describe los procesos de diseño, construcción e implementación de una red de sensores inalámbricos para la medición del consumo eléctrico en dispositivos de bajo voltaje. Este trabajo se divide principalmente en cuatro secciones. En primer lugar, se diseñan todos los circuitos de medición para voltaje y corriente. Estos circuitos son simulados utilizando el programa Orcad para visualizar su comportamiento. Posteriormente, se diseña la tarjeta de adquisición de datos PCB utilizando el programa Altium Designer. Esta tarjeta contiene un circuito para la medición del voltaje y cinco circuitos para la medición de corriente. En segundo lugar, se implementa el procesamiento digital y la transmisión inalámbrica de los datos utilizando la tarjeta EM430F6137RF900. La programación de dicho procesamiento y transmisión se realiza en el programa Code Composer Studio utilizando lenguaje C y Assembler. Para la recepción de datos se utiliza la tarjeta CC1111 USB Dongle programada en lenguaje C utilizando el programa IAR Embedded Workbench IDE for 8051. En tercer lugar, todas las mediciones realizadas se presentan en una interfaz gráfica [GUI] implementada en el programa LabVIEW. Esta interfaz permite la exportación de dichas mediciones en un archivo [.csv]. Finalmente, se realiza un análisis estadístico de los resultados obtenidos. Este análisis sugiere la implementación de una función de corrección basada en una regresión lineal de los datos para conseguir el correcto desempeño del sistema de medición propuesto.

Palabras Clave: Red de sensores inalámbricos, tarjeta de adquisición de datos, encapsulamiento de datos, comunicación inalámbrica, transmisión USB-Serial.

Abstract

This thesis describes the process of design, construction and implementation of a wireless sensor network designed for measuring electric consumption of low voltage devices. This work is mainly divided into four sections. In the first one, all measurement circuits for current and voltage are designed. These circuits are simulated, using the software Orcad, to visualize their behavior. Then, the printed circuit board PCB is designed using Altium Designer software. This PCB contains a circuit for measuring voltage and five circuits for measuring current. In the second section, digital processing and wireless data transmission is implemented using the EM430F6137RF900. The programming of such processing and transmission processes are performed in the Code Composer Studio software using languages C and Assembler. For data receiving, the CC1111 USB Dongle is used. This device is programmed in C language using the 'IAR Embedded Workbench IDE for 8051' software. In the third section, all measurements made are presented in a graphical user interface [GUI] implemented in LabVIEW. This interface allows the export of these measurements in a [.csv] file. Finally, in the last section, a statistical analysis of the results is performed. This analysis suggests the implementation of a correction function based on a linear regression of the data which allows a proper performance of the proposed measurement system.

Keywords: Wireless sensor network, data acquisition, printed circuit board, data encapsulation, wireless communication, USB-Serial transmission.

TABLA DE CONTENIDO

Resumen	7
Abstract	8
Capítulo 1:	14
Objetivos	14
Antecedentes	15
Justificación	16
Metodología	17
Revisión Bibliográfica	18
Red de Sensores Inalámbricos (RSI)	19
Medición de consumo eléctrico con sistemas RSI	21
Hacia los sistemas de respuesta a la demanda	23
Capítulo 2:	24
Introducción	24
Metodologías de Medición	28
Hardware de medición: Microprocesadores	30
Software de medición: Interfaces GUI, Programación de alto y bajo nivel	31
Fundamentos sobre la transmisión inalámbrica	32
Capítulo 3:	36
Diseño e implementación de la tarjeta de adquisición de datos	36
Familia MPS430 y CC430RF61xx Texas Instruments	37
Sistema de Reloj Unificado (SRU) – Unified Clock System (UCS)	39
Módulos I/O Digitales – Digital I/O Module	41
Temporizador_A – Timer_A (TAx)	42
Reloj de tiempo real (RTC)	44
Multiplicador de Hardware de 32-bits – 32-Bit Hardware Multiplier (MPY32)	45
ADC12_A	46
Limitaciones eléctricas CC430F6137	48
Circuitos de medición y acondicionamiento	49
Diseño, funcionamiento y simulación del sensor de voltaje	49
Diseño, funcionamiento y simulación del sensor de corriente	53

Diseño y planificación de la tarjeta PCB (Printed Circuit Board)	58
Modelamiento tridimensional del sistema de adquisición de datos	61
Capítulo 4:	64
Implementación de Software	64
Programación y procesamiento digital de las señales	64
Rutinas y algoritmos	65
Ficheros o archivos de cabecera (header file).h	67
Inicialización del sistema (emeter-setup.c)	68
Proceso de background del sistema (background.c)	72
Procesamiento y cálculo de variables	74
Corrección de fase: Filtro FIR	76
Proceso de foreground del sistema (foreground.c)	77
Lazo principal (main) del sistema (main.c)	78
Parámetros de calibración	79
Capítulo 5:	80
Fundamentos de la red implementada	80
Estructura y protocolos de comunicación implementados	80
Modelo de transmisión	80
Protocolo de acceso al medio (MAC)	81
Protocolo de enrutamiento	82
Módulo de radio frecuencia CC1101 (RF1A)	82
Manejo y funcionamiento del módulo RF1A	84
Fundamentos operativos del núcleo RF1A en la red inalámbrica implementada	86
Programación de la red inalámbrica en banda ISM	89
Rutinas y algoritmos	89
Ficheros o archivos de cabecera (header file).h	90
Módulo RF1A dentro de la inicialización del sistema (emeter-setup.c)	91
Módulo RF1A dentro del proceso de foreground del sistema (foreground.c)	93
Módulo RF1A dentro del lazo principal (main) del sistema (main.c)	94
Capítulo 6:	97
Fundamentos en la recepción y presentación de datos	97
CC1111 USB Dongle	97
Hardware de Recepción	98
Controlador USB	98

Software de Recepción	102
Interfaz Gráfica de Usuario [GUI]	109
Desencapsulamiento de la información	110
Implementación de la GUI utilizando NI LabVIEW	110
Capítulo 7:.....	113
Estructura física resultante del sistema DAQ	113
Resultados del funcionamiento del sistema DAQ implementado	115
Análisis del funcionamiento del sistema DAQ implementado	118
Factores estadísticos de corrección	119
Resultados del funcionamiento del sistema DAQ con funciones de corrección	122
Análisis del funcionamiento del sistema DAQ con funciones de corrección	123
Capítulo 8:.....	125
Conclusiones	125
Recomendaciones	128
Referencias Bibliográficas	129
ANEXOS:	133
ANEXO A 1.1: Esquemático de conexión EM430F6137RF900	133
ANEXO A 1.2: Diagrama PCB EM430F6137RF900	134
ANEXO B 1.1: Diagrama de bloques módulo ADC12_A	135
ANEXO B 1.2: Diagrama de bloques modo autoscan	136
ANEXO C 1.1: Simulación del sensor de voltaje	137
ANEXO C 1.2: Diagrama de bloques ACS712-20A-T	138
ANEXO C 1.3: Script en MATLAB para el dimensionamiento del sensor de corriente ...	139
ANEXO C 1.4: Simulación del sensor de corriente	140
ANEXO D 1.1: Esquemático Final PCB	141
ANEXO E 1.1: Algoritmo implementación filtro FIR en Matlab	142
ANEXO F 1.1: Mediciones por canal del sistema DAQ	144
ANEXO F 1.2: Pruebas t para las mediciones del sistema DAQ	146
ANEXO F 1.3: Gráficas de dispersión por canal del sistema DAQ	148
ANEXO F 1.4: Mediciones por canal del sistema DAQ con funciones de corrección	154
ANEXO F 1.5: Tablas de error para las mediciones del sistema DAQ con funciones de corrección	156

FIGURAS Y TABLAS

<i>Figura 1: Modelo OSI</i>	33
<i>Figura 2: Formato Paquete</i>	34
<i>Figura 3: TAx Up Mode</i>	42
<i>Figura 4: TAx Out Modes</i>	43
<i>Figura 5: ADC12_A Pulse Mode</i>	47
<i>Figura 6: Circuito Sensor de Voltaje</i>	49
<i>Figura 7: Comportamiento ACS712-20A-T</i>	54
<i>Figura 8: Sensor de Corriente</i>	55
<i>Figura 9: Comportamiento Sensor Corriente</i>	58
<i>Figura 10: PCB (vista superior)</i>	60
<i>Figura 11: PCB (vista inferior)</i>	61
<i>Figura 12: Estructura tipo Regleta</i>	62
<i>Figura 13: Sistema de medición completo</i>	63
<i>Figura 14: Diagrama Bloques Sistema Medición</i>	66
<i>Figura 15: Flujograma Inicialización (setup)</i>	69
<i>Figura 16: Proceso de conversión y muestreo</i>	72
<i>Figura 17: Flujograma background</i>	73
<i>Figura 18: Flujograma de señal por canal</i>	74
<i>Figura 19: Interpolación lineal</i>	75
<i>Figura 20: Filtro FIR de orden cero</i>	76
<i>Figura 21: Flujograma Foreground</i>	77
<i>Figura 22: Flujograma main</i>	78
<i>Figura 23: Modelo HAL</i>	81
<i>Figura 24: Diagrama de Bloques Módulo RF1A</i>	83
<i>Figura 25: Comunicación en el módulo RF1A</i>	85
<i>Figura 26: Encapsulamiento Implementado</i>	86
<i>Figura 27: Flujograma Inicialización (setup) con módulo RF1A</i>	91
<i>Figura 28: Flujograma Foreground-RF1A</i>	94
<i>Figura 29: Flujograma main-RF1A</i>	95
<i>Figura 30: Controlador USB</i>	99
<i>Figura 31: Diagrama de Bloques Módulo RF</i>	101
<i>Figura 32: Flujograma del lazo principal rf_modem.c</i>	105
<i>Figura 33: Flujograma USB IN</i>	106
<i>Figura 34: Flujograma appReceiverTask ()</i>	108
<i>Figura 35: Flujograma CC1111 ISR</i>	109
<i>Figura 36: GUI</i>	111
<i>Figura 37: a) Modelo PCB b) Estructura PCB final (vista superior)</i>	113
<i>Figura 38: a) Modelo PCB b) Estructura PCB final (vista inferior)</i>	114
<i>Figura 39: a) Modelo PCB + RF b) Estructura PCB + RF</i>	114
<i>Figura 40: a) Modelo 3D Sistema DAQ b) Estructura Sistema DAQ</i>	115
<i>Figura 41: Diagrama Conexión de Prueba DAQ</i>	116
<i>Figura 42: Gráfica de dispersión voltaje Chan_1</i>	120
<i>Figura 43: Gráfica de dispersión corriente Chan_1</i>	121
<i>Figura 44: Gráfica de dispersión potencia activa Chan_1</i>	121
<i>Figura 45: Esquemático EM430F6137RF900</i>	133
<i>Figura 46: Diagrama PCB EM430F6137RF900</i>	134
<i>Figura 47: Módulo ADC12_A</i>	135

<i>Figura 48: Modo Autoscan</i>	136
<i>Figura 49: Simulación Sensor Voltaje</i>	137
<i>Figura 50: ACS712-20A-T</i>	138
<i>Figura 51: Simulación Sensor Corriente</i>	140
<i>Figura 52: Esquemático PCB</i>	141
<i>Figura 53: Gráfica de dispersión voltaje Chan_2</i>	148
<i>Figura 54: Gráfica de dispersión corriente Chan_2</i>	148
<i>Figura 55: Gráfica de dispersión potencia activa Chan_2</i>	149
<i>Figura 56: Gráfica de dispersión voltaje Chan_3</i>	149
<i>Figura 57: Gráfica de dispersión corriente Chan_3</i>	150
<i>Figura 58: Gráfica de dispersión potencia activa Chan_3</i>	150
<i>Figura 59: Gráfica de dispersión voltaje Chan_4</i>	151
<i>Figura 60: Gráfica de dispersión corriente Chan_4</i>	151
<i>Figura 61: Gráfica de dispersión potencia activa Chan_4</i>	152
<i>Figura 62: Gráfica de dispersión voltaje Chan_5</i>	152
<i>Figura 63: Gráfica de dispersión corriente Chan_5</i>	153
<i>Figura 64: Gráfica de dispersión potencia activa Chan_5</i>	153
<i>Tabla 1: Características CC430RF6137</i>	38
<i>Tabla 2: SRU</i>	40
<i>Tabla 3: Módulos I/O Digital</i>	41
<i>Tabla 4: Módulo MPY32</i>	45
<i>Tabla 5: Limitaciones Eléctricas</i>	48
<i>Tabla 6: Archivos de cabecera</i>	67
<i>Tabla 7: Características Módulo RF1A</i>	84
<i>Tabla 8: Archivos de cabecera módulo RF1A</i>	90
<i>Tabla 9: Configuración módulo RF1A</i>	92
<i>Tabla 10: Características Módulo RF</i>	102
<i>Tabla 11: Módulo RF C1111F32</i>	107
<i>Tabla 12: Características GUI</i>	112
<i>Tabla 13: Mediciones Canal_1 (promedios)</i>	117
<i>Tabla 14: Mediciones Canal 1 con corrección (promedios)</i>	122
<i>Tabla 15: Tabla de error Canal 1</i>	123
<i>Tabla 16: Mediciones Canal_2 (promedios)</i>	144
<i>Tabla 17: Mediciones Canal_3 (promedios)</i>	144
<i>Tabla 18: Mediciones Canal_4 (promedios)</i>	145
<i>Tabla 19: Mediciones Canal_5 (promedios)</i>	145
<i>Tabla 20: Pruebas t-pareadas</i>	146
<i>Tabla 21: Pruebas t-de una muestra</i>	147
<i>Tabla 22: Mediciones Canal_2 con corrección (promedios)</i>	154
<i>Tabla 23: Mediciones Canal_3 con corrección (promedios)</i>	154
<i>Tabla 24: Mediciones Canal_4 con corrección (promedios)</i>	155
<i>Tabla 25: Mediciones Canal_5 con corrección (promedios)</i>	155
<i>Tabla 26: Tabla de error Canal_2</i>	156
<i>Tabla 27: Tabla de error Canal_3</i>	156
<i>Tabla 28: Tabla de error Canal_4</i>	157
<i>Tabla 29: Tabla de error Canal_5</i>	157

Capítulo 1:

Introducción

Objetivos

General:

- Diseño y construcción de una red de sensores inalámbricos configurables para el monitoreo y medición del consumo eléctrico de dispositivos de bajo voltaje.

Específicos:

- Familiarización con la tecnología basada en redes de sensores inalámbricos y sus aplicaciones.
- Análisis de la arquitectura y funcionamiento de los microcontroladores de bajo consumo.
- Definición y propuesta de la estructura física de la red de sensores inalámbricos.
- Definición y propuesta de la estructura lógica de la red de sensores inalámbricos.
- Diseño y creación del sistema de medición eléctrica en cada nodo.
- Establecimiento de la interface RF sub-1 [GHz] (radio frecuencia), su diseño, alcance y limitación.
- Desarrollo de una interfaz gráfica para la visualización y monitoreo de los datos medidos dentro de la red.
- Evaluación de la red desarrollada y análisis sobre el desempeño de la misma.

Antecedentes

El crecimiento continuo y el uso masivo que tienen hoy en día las distintas aplicaciones de bajo voltaje, han impulsado la investigación y el desarrollo de nuevas tecnologías diseñadas para analizar y controlar el consumo eléctrico del hogar. Sin embargo, a pesar de que varias han sido las iniciativas creadas con el propósito de medir este consumo, no todas han resultado económicamente convenientes o simplemente no han sido eficientes. Cabe recalcar que cuando se menciona el término “económico” no solo hace referencia al aspecto monetario sino también a términos energéticos. De acuerdo con Hanssen (2014), existe una gran tendencia en los instrumentos desarrollados para calcular el consumo de energía eléctrica de arrojar un resultado mayor al consumo real. Por esta razón, se han creado varios modelos con el objetivo de estandarizar la medición de energía y reducir la diferencia entre la energía calculada y la medida real. Ejemplo de ello, es la utilización de sensores de medición de energía eléctrica interconectados por señales inalámbricas.

Al hablar de sistemas de medición inalámbrica, los sistemas RSI [Red de sensores inalámbricos] son una alternativa viable. Éstos forman parte de un desarrollo tecnológico moderno, el cual nace de la necesidad de crear sistemas de medición de baja potencia con una distribución espacial adaptable. Los sistemas RSI además de estar interconectados de forma inalámbrica, son estandarizados y utilizan usualmente una antena simple omnidireccional en cada nodo (Stankovic, 2006). Una de las aplicaciones de mayor relevancia dentro de los sistemas RSI es el monitoreo y control de la eficiencia energética de una unidad habitacional. Para lograr esto, es necesario la instalación de cada nodo de la red en cada dispositivo electrónico del hogar (Buratti, Conti, Dardari & Verdone, 2009).

Justificación

El aumento del consumo de energía eléctrica y, consecuentemente, la disminución de este recurso, es presentemente un tema de preocupación mundial. No obstante, en el Ecuador éste no presenta aún un amplio campo de desarrollo a pesar de que sí representa una emergente necesidad. El constante crecimiento del consumo eléctrico en los últimos años se debe principalmente al incremento de la cantidad de dispositivos electrónicos de consumo diario tales como celulares, tabletas y electrodomésticos. Debido a que impedir el uso de estos dispositivos en un país en vías de desarrollo significaría un retroceso para el mismo, resulta sumamente necesario buscar soluciones alternativas que den un alivio a la situación actual del sistema nacional de electricidad. Es por eso que la medición del consumo de energía en los hogares se ha transformado en un ámbito crucial de investigación en el país.

En la actualidad, se han planteado dos soluciones principales para este problema. Una primera propuesta es la creación de nuevas plantas de generación de energía eléctrica que provean de apoyo secundario al momento de un incremento atípico en la demanda. Debido al alto costo de esta solución, se ha empezado a investigar y desarrollar una segunda propuesta, la misma que consiste en métodos o sistemas que puedan, de cierta forma, monitorear y controlar el consumo eléctrico del hogar. “Los usuarios demandan dispositivos y sistemas con mejor funcionalidad. En estos sistemas, sensores son utilizados para proveer información acerca de los parámetros de medición y el estado del sistema” (Hac, 2003).

Los sistemas para el monitoreo del consumo eléctrico de bajo voltaje mediante una red de sensores inalámbricos presentan una alternativa viable al problema del aumento en

la demanda de energía eléctrica del país por varias razones. En primer lugar, estos sistemas son más económicos que una planta de generación. Los RSI presentan una ventaja sobre los sistemas de comunicación guiada ya que el medio de transmisión es el aire. En segundo lugar, con estos sistemas se pueden beneficiar tanto al consumidor como al proveedor ya que tras una correcta implementación, se puede conseguir controlar de manera automática el consumo del hogar. Por ejemplo, en el caso de tener una tarifa variable por consumo eléctrico, con estos sistemas se puede llegar a optimizar dicho consumo, y cortar con ello los picos de tarifa máxima, lo cual resulta conveniente para el consumidor. Además, se disminuiría notoriamente la demanda de energía en las horas pico del día y por lo tanto, el costo que representaría para el proveedor cubrir este incremento en la demanda.

Es importante recalcar que los RSI también pueden presentar inconvenientes y dificultades. La ventaja es que en este caso, al igual que con cualquier sistema de medición en base a comunicación inalámbrica, estos problemas pueden ser tratados, mejorados y optimizados a través de un correcto diseño e implementación del sistema. De manera particular, los autores Sohraby, Minoli y Znati (2007) mencionan que la utilización de sistemas RSI para la medición del consumo eléctrico del hogar puede resultar en la disminución del consumo mensual equivalente a un ahorro anual que, en Estados Unidos, es igual a cincuenta y cinco billones de dólares. Por lo tanto, es fundamental la investigación sobre este tema.

Metodología

- a. Formación de una base teórica del proyecto: Explicar qué son las redes de sensores inalámbricos, exponer los fundamentos sobre la medición de energía eléctrica, la arquitectura de los microcontroladores, la programación de alto y bajo nivel y las

características y protocolos fundamentales para la transmisión de datos por radio frecuencia (RF).

- b. Diseño y simulación de los circuitos para la medición del voltaje y corriente en la carga.
- c. Planificación del material físico (tarjeta de adquisición de datos) utilizada para las mediciones necesarias a través del diseño digital del mismo.
- d. Modelamiento físico y estructural del sistema de medición a través de un diseño tridimensional.
- e. Implementación del modelo tridimensional del sistema de medición en conjunto con la tarjeta de adquisición de datos.
- f. Programación para la obtención, procesamiento y transmisión de datos medidos a través de varios algoritmos implementados.
- g. Implementación de una interfaz (GUI) para el posterior análisis de datos y evaluación de la red implementada.
- h. Evaluación general del rendimiento de la red (RSI) utilizando los resultados obtenidos de cada uno de los medidores de la red.
- i. Presentación de los resultados obtenidos.
- j. Elaboración de las conclusiones y recomendaciones.

Revisión Bibliográfica

Como se ha expuesto ya, el constante crecimiento en el consumo de energía eléctrica ha impulsado el desarrollo e investigación de varias tecnologías para la medición de energía eléctrica. Algunas de las tecnologías más efectivas están relacionadas con sensores interconectados por señales inalámbricas que tienen por objetivo analizar y optimizar el consumo eléctrico del hogar. En la presente sección de este trabajo se presenta una breve revisión bibliográfica que explica las principales características de la tecnología de red de sensores inalámbricos [RSI]. Como complemento, se resumen también a

continuación algunas investigaciones sobre la aplicación de estos sistemas en la medición del consumo eléctrico como parte de un sistema de respuesta a la demanda [DSM].

Red de Sensores Inalámbricos (RSI)

Una red de sensores inalámbricos puede ser descrita como una red de nodos los cuales establecen diversos ambientes de interacción entre la variable real medida y el usuario o interfaz desarrollada (Buratti, Conti, Dardari y Verdone, 2009). A través del empleo de esta red, lo que se busca es la recolección de mediciones de uno o más parámetros físicos, los cuales pueden ser transmitidos sin necesidad de una conexión directa por cable a un concentrador principal para su posterior procesamiento y análisis. Según indica la descripción de los autores Sohraby, Minoli & Znati (2007), una RSI se encuentra principalmente compuesta de una distribución de sensores, una red inalámbrica de conexión, un concentrador central de información y una central final para el procesamiento de datos o señales. En este contexto, se puede observar que una RSI, además de poseer varios dispositivos físicos, abarca tanto un conjunto de parámetros de diseño tal como protocolos de acceso al medio y protocolos de enrutamiento, así como la finalidad o aplicación deseada para el proyecto en desarrollo.

En los últimos años, varias investigaciones en base a los sistemas RSI han logrado desarrollar nuevos métodos o protocolos de manejo de datos dentro de la red. A partir de los resultados expuestos, se ha llegado a la conclusión de que para los sistemas RSI se necesita el establecimiento de un protocolo MAC entre los nodos de los sensores para el acceso compartido al medio (Sohraby et al., 2007). Esto se debe a que a pesar de que el medio de comunicación inalámbrico sea el aire, este también puede presentar problemas de congestión o de saturación. Los protocolos MAC se convierten en este sentido en

“oficiales de tránsito” de las redes inalámbricas. Los protocolos de acceso al medio más comunes para todo sistema RSI pueden ser de asignación fija como el TDMA y el CDMA, de asignación bajo demanda como el método de reservación, o de asignación aleatoria como los protocolos ALOHA y CSMA (Sohraby et al., 2007).

En todo tipo de red en la cual se encuentren varios nodos de transmisión, establecer reglas del uso del medio es fundamental para el correcto funcionamiento del mismo. Con la implementación de dichas reglas, se pretende optimizar varios parámetros de funcionamiento del sistema tales como el retardo, el tiempo de transmisión, la escalabilidad y la estabilidad. Estos dos últimos parámetros, son de suma importancia en todo sistema RSI ya que la escalabilidad permite al sistema funcionar de la misma manera con el mismo desempeño a pesar del aumento de los nodos o puntos de conexión en el sistema; y la estabilidad por su parte, permite al sistema de comunicación manejar fluctuaciones de tráfico por largos períodos de tiempo.

Una de las principales características de todo sistema RSI es el protocolo de enrutamiento. Un protocolo de enrutamiento consiste en un conjunto de reglas directivas para la correcta transmisión de datos de un sensor a otro. El protocolo de enrutamiento comúnmente se escoge convenientemente dependiendo de la aplicación y lo que se requiere realizar con un sistema RSI. Sin embargo, otro método para identificar el protocolo adecuado consiste en definir la topología de los sensores en la red RSI (Mishra & Kaur, 2014). Si la intención es ver al sistema de comunicación RSI en un modelo de capas, los protocolos de enrutamiento se encuentran en la capa de red, la cual, como expone Hac (2003), mantiene dos funciones principales: direccionamiento de nodos y enrutamiento.

Como resultado de varias investigaciones realizadas se han establecido dos grupos de protocolos de enrutamiento, mismos que pueden ser de tipo proactivo o de tipo reactivo. Por un lado, los protocolos proactivos mantienen actualizada la tabla de enrutamiento en cada nodo de la red. Los protocolos de vector distancia y de estado de enlace son ejemplos de protocolos proactivos. Por otro lado, los protocolos reactivos crean actualizaciones de enrutamiento bajo demanda. Vector distancia bajo demanda y ruteo de fuente dinámica son ejemplos de protocolos reactivos. Para poder determinar el protocolo adecuado, se puede considerar la propuesta de los autores Lin, Jianming, Hongzhou & Zhiyong (2014). La investigación de estos autores permite concluir que los protocolos de enrutamiento reactivos presentan una superioridad experimental que depende del factor de conexión analizado, el cual se convierte a su vez, en ciertos casos, en el determinante para seleccionar el protocolo final.

A pesar de todos los protocolos de enrutamiento desarrollados, existen nuevas alternativas para la optimización de sistemas RSI. Por ejemplo, el resultado de los autores Nokhanji & Hanapi (2014), exhibe un sistema de enrutamiento en paralelo basado en un algoritmo LEACH. Esta nueva alternativa, muestra una optimización en el desempeño del sistema RSI. Sin embargo, es importante recalcar que todo sistema RSI bajo cualquier protocolo de enrutamiento presenta limitaciones relacionadas con parámetros físicos o de red. Ejemplo de ello son la seguridad de datos, la congestión del medio y la respuesta en tiempo real (Stankovic, 2006).

Medición de consumo eléctrico con sistemas RSI.

Como se ha mencionado ya, tanto de la creciente preocupación global por disminuir el consumo de energía eléctrica como de la necesidad de reducir la discrepancia

que tienen los instrumentos de medición entre el valor calculado con el valor medido, ha surgido la creación de varios modelos de medición de consumo eléctrico con el objetivo de estandarizar la medición de energía. El primer modelo de consumo que se propone se basa en la medición del perfil actual de carga de los hogares y su relación estadística con variables de entrada como es el tiempo de medición. El segundo modelo de consumo se basa en cambio, en los niveles de perfil pertenecientes a datos de uso final de energía del consumidor (Svehla, 2011). Existe una posible interrelación entre los sistemas RSI con los modelos de medición del consumo energético del hogar. Se debe recordar que los sensores pertenecientes a los sistemas RSI, al igual que estos modelos, pueden medir variables físicas, entre ellas, las variables eléctricas.

Según Buratti et al. (2009) una de las aplicaciones de mayor relevancia dentro de los sistemas RSI se basa en el monitoreo y control de la eficiencia energética de una casa. Para lograr esto, es necesaria la instalación de cada nodo de la red en cada dispositivo electrónico del hogar. La recolección de datos en cada nodo de la red se basa en parámetros como la frecuencia de medición, el período de medición, entre otros. Acorde a Svehla (2011), la recolección de datos de este tipo se debe medir con un rango de frecuencias entre una hora a un minuto dentro de un período de carácter mensual. Utilizando el esquema de medición propuesto por esta autora, se logra formar una extensa red de sensores que monitorean y recolectan la información del consumo eléctrico en cada nodo. Es importante resaltar que en sistemas RSI se utilizan algoritmos jerárquicos de tipo SMDC para la recolección de datos (Lasheng, Jie & Renjie, 2013).

Hacia los sistemas de respuesta a la demanda

Uno de los objetivos principales de la medición y la optimización del consumo de energía eléctrica a través de sistemas de medición RSI es la creación del vínculo entre la vivienda y la grilla o red inteligente de distribución eléctrica. Al incrementar sistemas de medición del consumo energético, se puede crear mecanismos para manejar la demanda de energía eléctrica de los consumidores en respuesta al suministro de energía. Según los autores Kiliccote, Piette & Ghatikar (2011) la respuesta a la demanda son justamente estos mecanismos, los cuales se han convertido en una tendencia contemporánea. De hecho, el crecimiento de esta tendencia refleja la necesidad actual del consumidor de desarrollar estándares para vincular su hogar o vivienda dentro de las redes inteligentes de distribución eléctrica (Holmberg, 2011).

Para concluir, el desarrollo de un sistema RSI para la medición del consumo energético representa un reto de implementación ya que éste debe considerar todos los parámetros de enrutamiento y acceso a medios expuestos en la sección anterior. Es por eso que este proyecto tiene por objetivo el desarrollo de un método eficiente de medición de energía eléctrica. Se pretende que en un futuro el consumidor final se vea beneficiado al integrar su hogar a una red de distribución controlada. Iniciar un consumo de energía eficiente, un consumo de energía inteligente, además de ser un aporte al medio ambiente, tendrá a la disminución en la tarifa mensual de consumo eléctrico en los hogares como una consecuencia positiva.

Capítulo 2:

Fundamentos Teóricos

Introducción

Uno de los principales objetivos de un sistema de medición eléctrica radica en su capacidad de cuantificación de las principales variables eléctricas. Las primeras variables esenciales que debe medir un sistema eléctrico son la corriente y el voltaje. A partir de la relación entre estas dos variables surge una tercera variable y una de las principales definiciones en todo sistema eléctrico: la potencia. La potencia es la cantidad de energía que absorbe una carga y se encuentra compuesta de una parte real denominada potencia activa en unidades [W] y una parte imaginaria conocida como potencia reactiva en unidades [VAr]. Finalmente, la suma de los cuadrados de cada uno de estos componentes resulta en la potencia aparente en unidades [VA]. Existe también una cuarta potencia, conocida como potencia distorsionante cuya magnitud es significativa el momento en que se analizan cargas no lineales.

Es importante recalcar que si el objetivo de un sistema de medición eléctrico radica en la cuantificación tanto de la corriente como del voltaje, es imperativa la preservación de la forma de onda para cada una de estas variables. En el caso de un sistema alterno, tanto el voltaje como la corriente se pueden representar como ondas cosenoidales como muestran las ecuaciones (1) y (2) respectivamente:

$$v = V_m \cos(\omega t) [V] \quad (1)$$

$$i = I_m \cos(\omega t) [A] \quad (2)$$

En este caso V_m e I_m representan los valores pico de la onda, ω representa la frecuencia de la onda y t es el tiempo.

Debido a que el sistema de distribución eléctrico occidental actual consiste en un sistema alterno, es importante para el lector reconocer algunas de las definiciones que surgen de este tipo de sistemas. Tres de estas definiciones que se exponen en esta sección son el voltaje y la corriente efectiva y la potencia. El valor efectivo tanto para el voltaje como para la corriente es la medida de la efectividad en la transmisión de potencia de cada una de estas variables hacia la carga. Asumiendo una representación periódica tanto del voltaje (v) como de la corriente (i) como se muestra en las ecuaciones (1) y (2), se puede definir matemáticamente a la corriente efectiva de la forma en que se presenta en la ecuación (3):

$$I_{rms} = \sqrt{\frac{1}{T} \int_0^T i^2 dt} \text{ [A]} \quad (3)$$

Aquí T es el período de la onda. De igual forma, el voltaje efectivo queda matemáticamente definido como se muestra en la ecuación (4):

$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T v^2 dt} \text{ [V]} \quad (4)$$

Por último, si se utiliza la definición de voltaje y corriente periódica previamente especificada y se la reemplaza en las definiciones anteriores, se obtiene una relación entre el valor efectivo de una onda periódica con su valor máximo como se muestra en la ecuación (5):

$$X_{rms} = \frac{X_m}{\sqrt{2}} \quad (5)$$

Aquí X_m representa el valor pico o máximo de la onda.

De manera general se definió a la potencia como la medida de la energía absorbida por una carga puntual. Ahora bien, independientemente del tipo de carga utilizada en un sistema de medición eléctrica, se pueden identificar cuatro tipos de potencia: una potencia real activa o promedio (P), una potencia reactiva o imaginaria (Q), una potencia de distorsión (D) y una potencia aparente ($|S|$). Esta última es el módulo de la potencia compleja (S) resultante de la suma [vectorial] de la potencia real más la imaginaria. A la potencia compleja transmitida a un elemento en su representación fasorial se la exhibe como en la ecuación (6):

$$S = \frac{vi^*}{2} = \frac{V_m I_m}{2} \angle \theta_v - \theta_i \text{ [VA]} \quad (6)$$

A partir de esta última definición, del resultado obtenido en la ecuación (5) y tomando en cuenta que la potencia compleja (S) es la suma de una potencia real más una imaginaria, se puede definir matemáticamente a la potencia aparente, la potencia activa y la potencia reactiva como se presenta en las ecuaciones (7), (8) y (9) respectivamente:

$$|S| = \frac{V_m I_m}{2} = V_{rms} I_{rms} \text{ [VA]} \quad (7)$$

$$P = \frac{V_m I_m}{2} \cos(\theta_v - \theta_i) \text{ [W]} \quad (8)$$

$$Q = \frac{V_m I_m}{2} \sin(\theta_v - \theta_i) \text{ [VAr]} \quad (9)$$

Es importante mencionar que todas estas definiciones son correctas siempre y cuando al voltaje y a la corriente se los represente de forma cosenoidal.

El momento en el que el voltaje o la corriente tienen una deformación con respecto a su forma cosenoidal original, se clasifica a esta señal como distorsionada. Esta distorsión, puede ser consecuencia de efectos transitorios como la conmutación de los capacitores, cortocircuitos en la red, entre otros. De igual forma, puede ser resultado de efectos permanentes como el de las armónicas en estado estable de las señales. Debido a la importancia de esta distorsión en la medición eléctrica, se define a la potencia distorsionante (D) como se presenta en la ecuación (10). Sin embargo, se puede expresar también a esta potencia a partir de las definiciones ya presentadas en su forma vectorial como se muestra en la ecuación (11).

$$D = V_{rms} \left(\sqrt{\sum_{h=2}^{\infty} I_h^2} \right) [VA] \quad (10)$$

$$D = |S| - \left(\vec{P} + \vec{Q} \right) [VA] \quad (11)$$

Aquí I_h representa la magnitud del componente armónico individual en la onda de corriente. Finalmente, se define al factor de potencia como se representa en la ecuación (12).

$$fp = \cos(\theta_v - \theta_i) \quad (12)$$

Al ángulo de factor de potencia ($\theta_v - \theta_i$) se lo puede caracterizar de dos formas: en adelanto cuando $(\theta_v - \theta_i) < 0$, o en retardo cuando $(\theta_v - \theta_i) > 0$. En el caso de que el ángulo de factor de potencia sea igual a cero, se deduce la presencia de una carga totalmente resistiva [carga reactiva igual a cero] dentro del sistema de medición.

Metodologías de Medición

Una vez que han sido aclarados los conceptos sobre las diferentes variables de medición, es prioritario definir los procedimientos a seguir para lograr medir de forma automática cada una de las variables previamente mencionadas. Las definiciones de las variables fundamentales, como el voltaje y la corriente, se realizaron bajo un esquema de tiempo continuo, sin embargo, la mayoría de sistemas digitales de medición son basados en la conversión análogo-digital. Por esta razón, se desarrolla una equivalencia discreta de cada una de las ecuaciones ya mencionadas.

Independientemente de la forma en la cual el diseñador consiga acondicionar las señales alternas tanto de voltaje como de corriente para su posterior conversión digital, el cálculo de las distintas variables se puede realizar como se indica a continuación. Para el caso del valor efectivo matemáticamente definido en las ecuaciones (3) y (4), la equivalencia discreta se presenta en las ecuaciones (13) y (14) respectivamente:

$$I_{rms} = SF \times \sqrt{\frac{\sum_{n=1}^{samples} I^2[n]}{samples}} [A] \quad (13)$$

$$V_{rms} = SF \times \sqrt{\frac{\sum_{n=1}^{samples} V^2[n]}{samples}} [V] \quad (14)$$

Aquí las variables utilizadas son: samples que es el número de muestras tomadas en un período de medición determinado, I que es el vector o arreglo de almacenamiento de datos transformados para la corriente de entrada, V que es el vector o arreglo de almacenamiento de datos transformados para el voltaje de entrada, y SF que es un factor de escalamiento.

Éste último es dependiente del circuito de acondicionamiento utilizado y del procesamiento digital de datos.

Para el caso de la potencia también existe una representación discreta de las ecuaciones ya mencionadas. La potencia activa o real se la puede calcular numéricamente de la forma en que se muestra en la ecuación (15):

$$P_{active} = (SF_V \times SF_I) \times \left(\frac{\sum_{n=1}^{samples} V[n] \times I[n]}{samples} \right) [W] \quad (15)$$

Aquí samples, I y V representan lo mismo que en las ecuaciones (13) y (14). De igual manera, SF_V y SF_I son los factores de escalamiento para voltaje y corriente dependientes del circuito de acondicionamiento utilizado y del procesamiento digital de datos.

Para el caso de potencia aparente, debido a que se asume una forma cosenoidal tanto para el voltaje como para la corriente, se puede calcular de forma discreta como se presenta en la siguiente ecuación (42):

$$P_{aparent} = (SF_V \times SF_I) \times V_{rms} \times I_{rms} [VA] \quad (16)$$

Aquí V_{rms} , I_{rms} son los valores efectivos discretos previamente definidos. Así mismo, debido a que la potencia reactiva representa mayor procesamiento digital de datos, para el correcto cálculo de la misma se debe insertar una fase de 90° en las muestras de voltaje. Es por eso que se recurre a la definición fundamental de potencia compleja para el cálculo de la potencia reactiva, como se muestra en la ecuación (17):

$$P_{reactive} = \sqrt{P_{aparent}^2 - P_{active}^2} [VAr] \quad (17)$$

Finalmente, el factor de potencia se lo puede calcular de forma discreta de la misma manera en que se ha mencionado en su definición en régimen continuo: bajo la relación o razón entre la potencia aparente con la potencia activa calculada.

Hardware de medición: Microprocesadores

Una vez establecidos los fundamentos teóricos de medición, es importante establecer brevemente los fundamentos teóricos del hardware que realiza dicha medición. En este caso, el proyecto de investigación que presenta esta tesis requiere de un procesamiento digital de las variables eléctricas que miden los sensores. De manera general, el procesamiento digital de las señales se realiza a través del uso y programación de microprocesadores. Los microprocesadores son sistemas digitales que permiten leer y ejecutar un set de instrucciones de lenguaje de máquina (Harris & Harris, 2013). A través de un correcto set de estas instrucciones, se pueden realizar varios procesos de análisis y manejo de datos; esto, siempre y cuando las señales de medición se encuentren dentro de los límites eléctricos del microcontrolador utilizado.

En la actualidad, son varias empresas las que se dedican al diseño y construcción de distintos microprocesadores. Muchos de ellos difieren en su arquitectura, rapidez, memoria, entre otras características. Por ejemplo, la familia de microprocesadores RF430 de la compañía Texas Instruments son dispositivos de bajo consumo de potencia y constan de un núcleo de radio frecuencia para distintas aplicaciones inalámbricas. Estos procesadores poseen varias opciones de procesamiento de datos como conversores análogo-digital, interfaces de comunicación serial, entre otros (Texas Instruments, 1998).

Software de medición: Interfaces GUI, Programación de alto y bajo nivel

Un lenguaje de programación es la combinación de un set de instrucciones en las cuales se detallan los procesos que se desea realizar con los datos o variables del sistema. Dentro de los lenguajes de programación, se tiene lenguajes de alto nivel como el lenguaje C y C++ y los lenguajes de bajo nivel como el lenguaje de máquina. Los lenguajes de alto nivel presentan por lo regular mayor entendimiento desde el punto de vista del programador. Los lenguajes de bajo nivel por su parte, son más utilizados por las máquinas ya que es el lenguaje binario que utiliza el procesador para ejecutar los comandos (Deitel & Deitel, 2009).

En la práctica, existen ciertos programas que nos permiten escribir varios sets de instrucciones en un lenguaje predeterminado y traducirlos a un lenguaje de máquina que depende del microprocesador utilizado. Estos programas, los cuales se conocen como compiladores, nos permiten programar y escribir combinando lenguajes de alto nivel con lenguajes de bajo nivel. Es importante saber diferenciar entre estos dos lenguajes de programación ya que ambos poseen notorias diferencias, desde la manera en que manejan sus variables y datos, hasta la forma de escritura que cada uno tiene. Recordemos que, mientras más bajo es el nivel de programación utilizado, más cercano es el manejo de variables a la arquitectura del microprocesador. Esta es la razón por la cual es fundamental primero reconocer el manejo de datos y registros del procesador utilizado antes de diseñar y escribir el set de instrucciones para el procesamiento digital de datos.

En la mayoría de sistemas de medición es importante el diseño de la forma en la cual el sistema muestra al usuario final los datos o resultados calculados. Es por eso que en muchos sistemas de medición se utilizan interfaces de usuario GUI, cuyas siglas en inglés

significan ‘Gráfica User Interface’. Los GUI, como su nombre lo indica, son interfaces gráficas que sirven para interactuar con las variables tanto de entrada como de salida de un sistema. En los GUI, el usuario puede visualizar los valores de las salidas así como alterar algún valor de entrada, mandar un mensaje al sistema, o incluso detener la operación del mismo. Estas interfaces se las puede programar en varios lenguajes de programación de alto nivel.

Fundamentos sobre la transmisión inalámbrica

En la presente sección se define y se describe los aspectos de mayor relevancia que caracterizan a la comunicación inalámbrica. Cada vez son más los sistemas electrónicos que dependen del medio inalámbrico para la transmisión y recepción de datos. Estos procesos pueden desarrollarse en una red de corto o mediano alcance como en una red Sub-1[GHz], ZigBee, BlueTooth o Wi-Fi [Estándar IEEE 802.11 b/g/n/ac], o en redes de largo alcance como las redes satelitales GPS. A pesar de la existencia de una gran variedad de redes o estándares de comunicación inalámbrica, todas estas alternativas poseen las mismas características y definiciones de estructura y diseño. Estas características y definiciones se detallarán a continuación.

El principal desafío de todo sistema de interconexión electrónico es la definición de la estructura o topología de red. Por esta razón, se han creado varios modelos o protocolos para optimizar y estandarizar la forma en que se transmite la información de un nodo a otro. En 1984, la organización internacional de estandarización (ISO) desarrolló un modelo de referencia para la interconexión de sistemas (OSI) compuesta por siete capas independientes una de otra como se muestra en la *Figura 1* (Cisco Systems, 2014). Todas estas capas describen el procesamiento realizado en los datos analizados tanto para la

transmisión como para la recepción. Para el caso de transmisión, los datos se procesan desde las capas de aplicación, es decir desde los programas de interfaz con el usuario, y se encapsulan en las capas de transporte para ser transmitidas en lenguaje de máquina por un medio físico o inalámbrico. Por su parte, el receptor recibe estas cápsulas de información, las abre, procesa la información contenida en ellas desde las capas de transporte y las transfiere a las capas de aplicación donde son analizadas y procesadas.

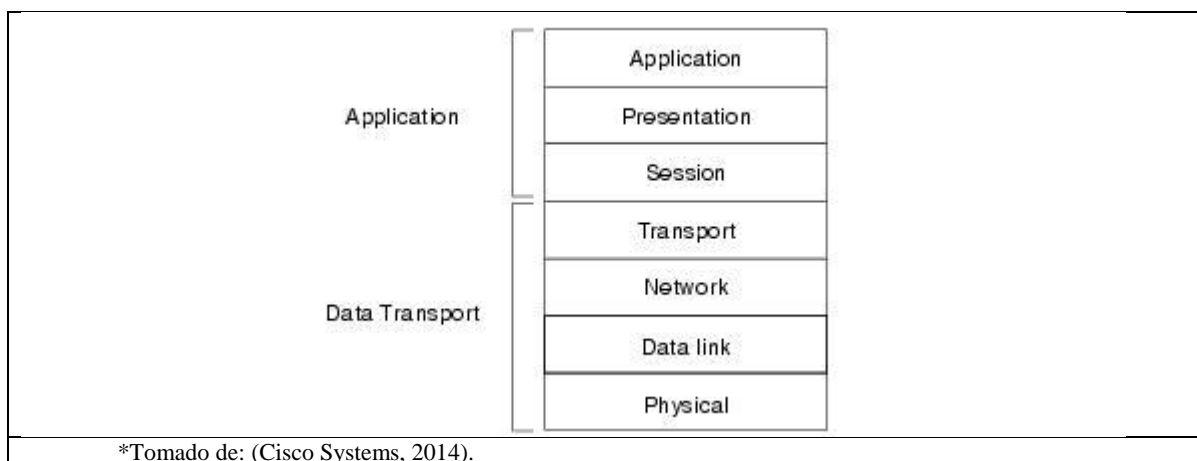


Figura 1: Modelo OSI

Para un correcto entendimiento de la comunicación inalámbrica dentro de los sistemas de interconexión electrónicos es necesario detallar el funcionamiento de las capas del modelo OSI, principalmente en la sección de transporte. La primera sección se compone de la capa física que es la encargada del procesamiento binario de las señales procesadas. Posteriormente, la segunda sección, la capa de estado de enlace, es la que provee un direccionamiento físico tanto de origen como de destino. En estas dos primeras capas, se inicia un proceso de encapsulamiento o empaquetamiento de información. Este proceso es fundamental en cualquier tipo de comunicación inalámbrica.

Los paquetes de información resultantes de este proceso, están compuestos principalmente de tres partes como se muestra en la *Figura 2*. Estas partes son:

- Cabeceras: contienen información de direccionamiento o sincronización.
- Información: cadena de bits que codifican la información transmitida.
- Cola: contiene información sobre el estado del enlace, indicador de fuerza de la señal recibida (RSSI) y, en muchos casos, una cadena de comprobación de la correcta recepción del paquete enviado (CRC).

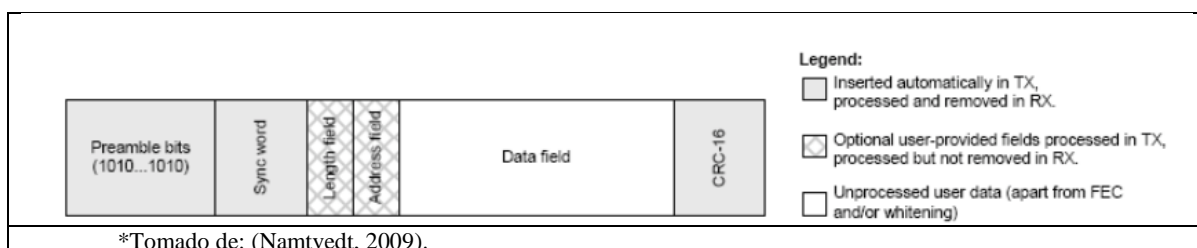


Figura 2: Formato Paquete

Todos estos paquetes, en la mayoría de los dispositivos de la compañía Texas Instruments, se almacenan en un buffer de tipo FIFO [“First In First Out”] de escritura o lectura continua, de memoria estática y de comunicación sincrónica o asincrónica en el momento de ser transmitidos o recibidos (Forstner, 1999).

Una vez culminado el proceso de encapsulamiento, en las dos últimas capas de esta sección, capa de red y capa de transporte, se definen los protocolos de enrutamiento y acceso al medio (MAC) para asegurar un debido desempeño de la red implementada. Por un lado, los protocolos de acceso al medio (MAC), establecen las reglas de comunicación entre los dispositivos dentro de la red y evitan así una posible saturación del medio, que en este caso es el aire. Por otro lado, los protocolos de enrutamiento envían información desde un sensor o nodo hasta el concentrador central. Entre los protocolos MAC se encuentra el

estándar IEEE 802.5 conocido también como Token Ring. Este protocolo, desarrollado en los años 70 por la compañía IBM, establece una topología en estrella de los integrantes dentro de la red. En el estándar Token Ring sólo el integrante que posee el “token” de la red es el único que tiene permiso para utilizar el medio y transmitir la información necesaria antes de pasar este “token” al siguiente integrante (Cisco Systems, 2012). Así mismo, un ejemplo de protocolo de enrutamiento en el caso del internet puede ser de tipo TCP o UDP.

La elección del protocolo de enrutamiento es fundamental para el desempeño de una RSI y varía en dependencia de un sistema multi-salto o de un solo salto (Sohraby et al., 2007). Algunas características físicas de la red también toman peso en la elección del protocolo de enrutamiento. Ejemplos de estas características son:

- Latencia: tiempo en el cual se transmite un paquete de datos desde su origen hasta su destino (Harris & Harris, 2013).
- Escalabilidad: capacidad de un sistema de mantener sus características de desempeño a pesar del número de dispositivos que se incrementen dentro de la red (Sohraby et al., 2007).
- Estabilidad: capacidad de un sistema de mantener sus características de desempeño a pesar de las distintas fluctuaciones en la carga del sistema (Sohraby et al., 2007).
- Resiliencia: capacidad de la red en auto repararse en caso de una falla.

Capítulo 3:

Sistema de Adquisición de Datos (DAQ)

Diseño e implementación de la tarjeta de adquisición de datos

Para una correcta implementación de un sistema de medición eléctrico, es necesario un dimensionamiento adecuado de todas las partes involucradas en la obtención, transformación, procesamiento y acondicionamiento de las distintas señales del sistema. En base a lo expuesto en el capítulo anterior, queda claro que si el objetivo principal de un sistema de medición eléctrico es la cuantificación de la potencia consumida por la carga en análisis, es fundamental un tratamiento para las señales tanto de voltaje como de corriente para que conserven en su totalidad la forma de onda. En este trabajo se expone detalladamente el proceso de elección de hardware, el dimensionamiento y la implementación del sistema de adquisición de datos propuesto para la construcción de un equipo que mide de forma independiente el consumo eléctrico hasta de cinco cargas diferentes. Este equipo, a través de un tratamiento tanto físico como digital de las señales de entrada, es capaz de calcular de forma automática el consumo eléctrico de la carga conectada y de algunas mediciones fundamentales tales como los valores efectivos de las ondas de entrada y el factor de potencia resultante.

En el presente capítulo se tratará únicamente el procedimiento físico de elaboración de la tarjeta de adquisición de datos dividido en tres partes. En las secciones a continuación se describen los parámetros de selección, características y diseño de cada uno de los elementos del sistema de medición eléctrico propuesto. En primer lugar, se describe la arquitectura del microcontrolador utilizado y se explican tanto sus características como

limitantes operativos para un correcto dimensionamiento dentro del sistema de medición. Posteriormente, tomando en cuenta todas las limitaciones eléctricas dentro del sistema, se procede al diseño, simulación e implementación de los sensores de voltaje y corriente utilizados para tratar y acondicionar dicha señal de entrada. Finalmente, una vez establecido el diseño eléctrico de medición, se procede a la construcción de una tarjeta PCB utilizando el programa Altium Designer.

Cabe reconocer que un correcto diseño eléctrico involucra todos sus componentes sin dejar de lado su apariencia, estructura física y dinámica mecánica. Por esta razón, después de finalizar, probar y evaluar todos los componentes eléctricos del sistema implementado, se realiza una propuesta tridimensional de la estructura física del mismo. Esta propuesta envuelve a todo este sistema de adquisición de datos en una arquitectura elegante, eficaz y eficiente tanto para el diseñador como para su usuario final. Por último, finalizada toda estructura física del sistema propuesto, se continúa en el siguiente capítulo con la estructura y procesamiento digital de las variables de medición resultantes de la tarjeta de adquisición de datos.

Familia MPS430 y CC430RF61xx Texas Instruments

Uno de los mayores retos al momento del diseño y planificación de todo sistema de medición es la elección de un microprocesador que se ajuste a todas las características del sistema requerido. Dentro de esta propuesta, se utilizó como microprocesador central al CC430RF6137 producido por Texas Instruments. Un resumen de las características de este chip se presenta en la *Tabla 1*. Este microprocesador a su vez forma parte de una tarjeta de evaluación de la serie EM430F6137RF900, la cual posee una estructura completa que permite el manejo adecuado de todos los puertos digitales además del uso del núcleo

inalámbrico descrito en la *Tabla 1*. Los diagramas de diseño PCB y esquemático de conexión para esta tarjeta de evaluación se los presentan en los anexos A1.1 y A1.2.

Tabla 1: Características CC430RF6137

Característica	Descripción
Voltaje de Entrada	Desde 1.8 [V] hasta 3.6 [V]
Arquitectura	Arquitectura tipo RISC de 16-bits con memoria extendida
Sistema y Periféricos	<ul style="list-style-type: none"> • Sistema de reloj unificado • Dos temporizadores de 16-bits con modalidad de captura o comparador • Dos interfaces para establecer comunicación serial • Conversor análogo digital con una resolución de 12-bits • Multiplicador de hardware de 32-bits
Núcleo inalámbrico de alto rendimiento Sub-1-GHz	<ul style="list-style-type: none"> • Velocidad de transmisión de datos hasta de 500 [kBaud] • Alta sensibilidad y bajo porcentaje de pérdida en paquetes de transmisión. • Soporte para modulación 2-FSK, 2-GFSK y MSK • Soporte para transmisiones tanto sincrónicas como asincrónicas

*Tomado de: SLAS554H (Texas Instruments, 2009a).

Una vez establecidos los parámetros generales del procesador a utilizar y la tarjeta de evaluación en la cual se encuentra inmerso, es necesario entrar en detalle sobre los

módulos, registros y configuraciones del microprocesador necesarios para que éste funcione bien. En las siguientes secciones, se describen los módulos necesarios para la implementación y configuración del procesador.

Sistema de Reloj Unificado (SRU) – Unified Clock System (UCS)

El sistema de reloj unificado [SRU] es el módulo que permite el manejo de las señales internas de reloj del microprocesador para la correcta sincronización o ejecución de las diferentes actividades solicitadas por otros módulos. El SRU soporta hasta cinco distintas fuentes de señal de reloj y suministra hasta tres distintas señales de reloj, tal como se resume en la *Tabla 2*. A pesar de la presencia de algunas fuentes internas de reloj, la arquitectura del procesador permite añadir una fuente externa a través del uso de un cristal para mayor estabilidad en las señales. Este cristal, conectado en los pines XIN y XOUT (ver esquemático en anexo A1.1) inicia desde el arranque del sistema con un valor elevado en su controlador para acelerar la estabilidad y convergencia del mismo. Posteriormente, el programador puede reducir vía software el valor de dicho controlador para disminuir el consumo de potencia del microcontrolador.

El módulo SRU posee también ciertas banderas contra protección de fallas en las diferentes fuentes de reloj. En el momento de arranque se enciende el cristal y si se detecta su malfuncionamiento, las banderas XT1LFOFFG, XT1HFOFFG y XT2OFFG son levantadas vía software hasta un reseteo del sistema o un correcto manejo de dichas banderas. En caso de que una falla sea detectada, si el cristal se encuentra configurado como fuente para cualquiera de las señales de reloj expuestas, dicha señal cambia de manera automática su fuente a DCOCLK para evitar interrumpir el funcionamiento del procesador.

Tabla 2: SRU

Fuentes de Reloj	
XT1CLK	Fuente de señal externa si se suelda en el espacio correspondiente un cristal de baja frecuencia (32768 [Hz]).
VLOCLK	Fuente de señal interna con una frecuencia nominal de 10 [kHz].
REFOCLK	Fuente de señal interna con una frecuencia típica de 32768 [Hz].
DCOCLK	Fuente de señal interna controlada digitalmente la cual puede ser estabilizada a través del uso del módulo FLL (Frequency Locked Loop).
XT2CLK	Fuente de señal externa proporcionada por un oscilador. Esta fuente es utilizada para el módulo de radio frecuencia.
Señales de Reloj	
ACLK	Señal de reloj auxiliar la cual puede ser seleccionada (vía software) como cualquiera de las cuatro primeras fuentes de reloj. Esta señal es utilizada por los módulos periféricos del procesador.
MCLK	Señal de reloj máster la cual puede ser seleccionada como cualquiera de las fuentes de reloj. Esta señal es utilizada por el CPU y el sistema.
SMCLK	Señal de reloj de subsistema la cual puede ser seleccionada como cualquiera de las fuentes de reloj. Esta señal es utilizada por los módulos periféricos del procesador.

*Tomado de: SLAU259E (Texas Instruments, 2009b).

Así mismo, el módulo SRU soporta un oscilador interno denominado MODOSC. Éste último puede ser utilizado de forma condicional o incondicional por diferentes módulos del procesador tales como el conversor análogo digital [ADC12_A]. Durante la conversión, dicho módulo puede realizar una solicitud incondicional que activa a MODOSC de manera temporal.

Módulos I/O Digitales – Digital I/O Module

Los módulos I/O digitales engloban a todos los puertos periféricos [P1 al P5 y PJ para el caso de CC430F6137] pertenecientes al microprocesador que pueden ser accedidos en bits o que pueden ser agrupados e interpretados como palabras o conjunto de bits. Para una correcta configuración de los mismos, se utilizan los siguientes registros presentados con su respectiva descripción en la *Tabla 3*.

Tabla 3: Módulos I/O Digital

Registros para los módulos I/O digitales

<i>PxIN</i>	<ul style="list-style-type: none"> • Bit = 0; Entrada en estado lógico bajo • Bit = 1; Entrada en estado lógico alto
<i>PxOUT</i>	<ul style="list-style-type: none"> • Bit = 0; Salida en estado lógico bajo • Bit = 1; Salida en estado lógico alto
<i>PxDIR</i>	<ul style="list-style-type: none"> • Bit = 0; Puerto configurado como entrada • Bit = 1; Puerto configurado como salida
<i>PxSEL</i>	<ul style="list-style-type: none"> • Bit = 0; Puerto configurado como módulo I/O • Bit = 1; Puerto configurado según función auxiliar asignada

*Tomado de: SLAU259E (Texas Instruments, 2009b).

Es importante mencionar que para el caso del procesador CC430F6137 la función auxiliar del puerto dos [P2.0 al P2.7] se encuentra ligada al módulo de conversión análogo digital debido a que estos periféricos actúan como las entradas análogas para realizar dicha conversión (ver esquemático del anexo A1.1).

Temporizador_A – Timer_A (TAx)

El módulo TAx es un temporizador o comparador de 16-bits capaz de soportar hasta siete registros de captura o comparación, cada uno con su capacidad de producir una interrupción en el sistema. Para el caso de configuración como módulo contador, un registro de conteo TAxR aumenta o disminuye según el modo de operación en cada cambio positivo de la señal de reloj, en la cual puede estar como fuente la señal ACLK o la señal SMCLK. Existen cuatro diferentes tipos de modos de operación: de parada, conteo ascendente, conteo continuo o conteo alterno [arriba/abajo]. Sin embargo, en esta sección simplemente se enfatiza en uno de ellos debido a que resulta de mayor relevancia.

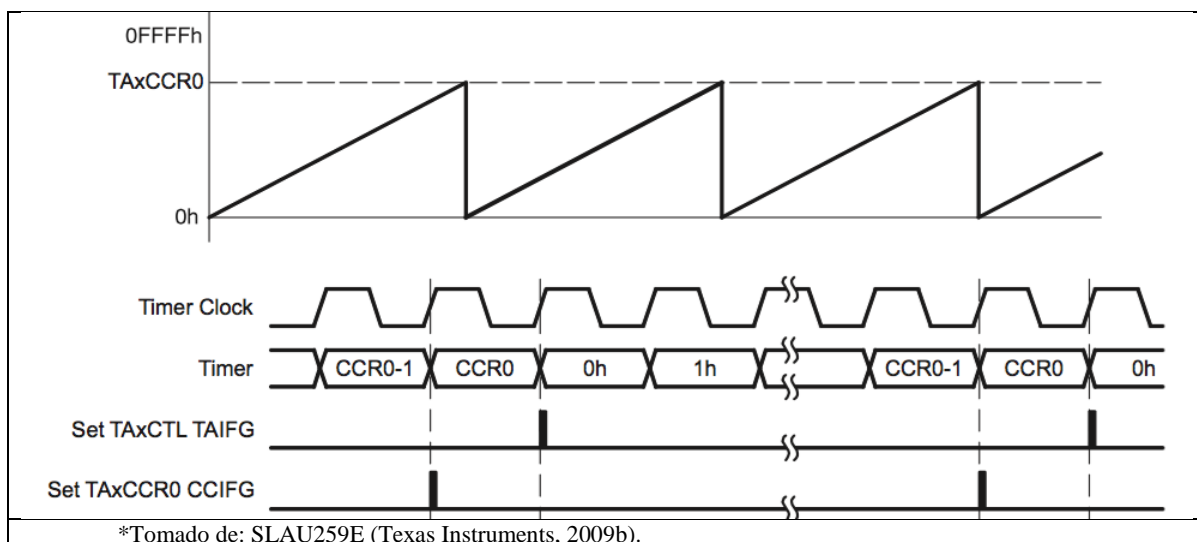


Figura 3: TAx Up Mode

En el modo de operación de conteo ascendente [Up mode], el registro de conteo inicia desde un valor igual a cero e incrementa hasta un valor definido por el registro TAxCCR0, el cual define el período de la interrupción y es distinto de su valor máximo (0xFFFF) hexadecimal. Si estas condiciones no sucedieran, se estaría configurando el módulo TAx en modo de operación continuo. El momento en el que el registro de conteo

alcanza su valor máximo (TAXCCR0), TAXR es reiniciado a su valor inicial y la bandera de interrupción TAXCCR0CCIFG es activada como se muestra en la *Figura 3*.

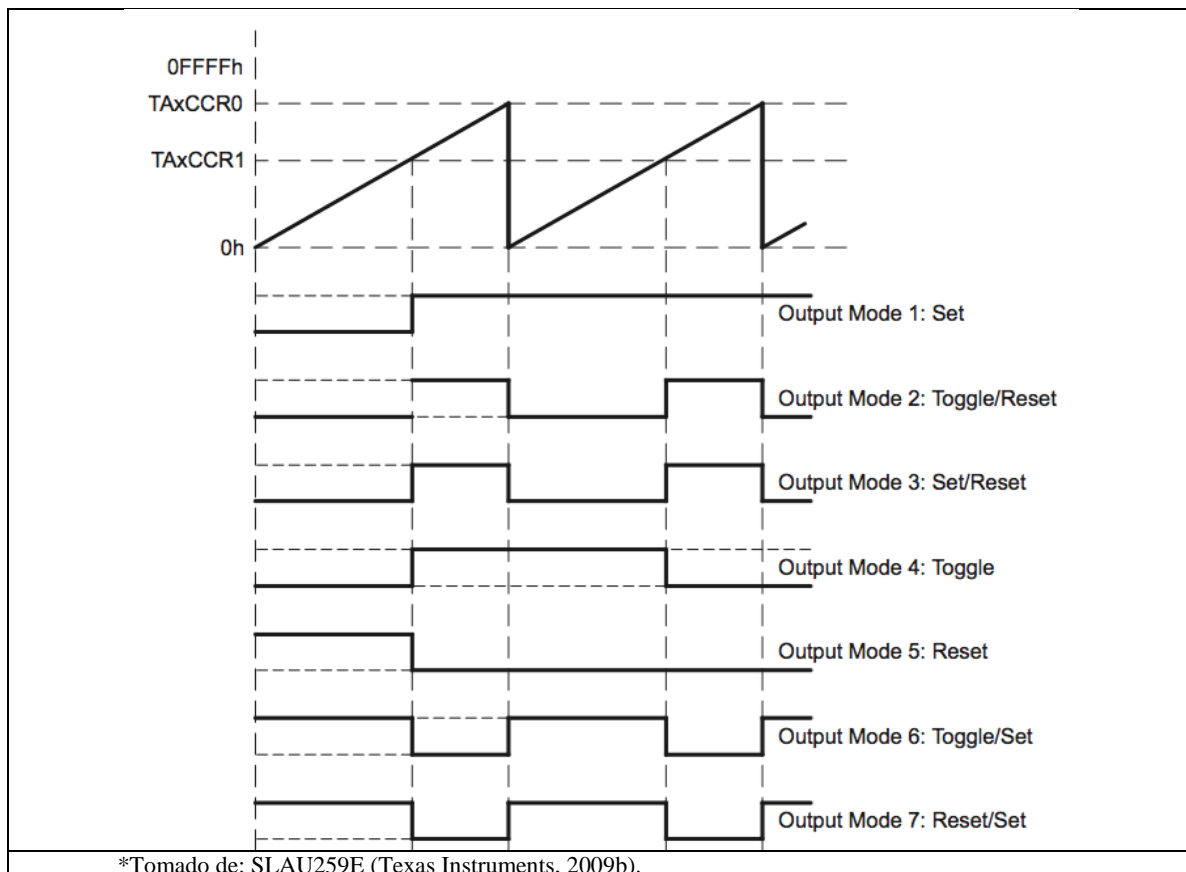


Figura 4: TAX Out Modes

Cada bloque de captura y comparación del módulo TAX contiene un registro de salida, el cual puede producir una señal en dependencia del modo de operación seleccionado. Como se puede observar en la *Figura 4*, esta señal de salida depende del modo de operación del temporizador al igual que del modo de operación de la salida como tal. En dependencia del valor que se configure para los registros de conteo [TAXCCR0 y TAXCCR1] se puede obtener una señal PWM controlada. Esta señal para el caso del procesador CC430F6137 puede vincularse con el módulo ADC_12 y convertirse en la señal que controle el tiempo de recolección y conversión de datos analógicos a digitales.

Reloj de tiempo real (RTC)

En la mayoría de aplicaciones de medición, es importante mostrar el dato medido junto a su estampilla de tiempo, es decir, el día, hora, minuto y segundo en el cual se realizó dicha medición. Por esta razón, es fundamental la utilización de un módulo que nos permita generar esta estampilla temporal en el sistema. El módulo RTC_A puede ser utilizado en modo calendario por medio de la utilización de registros individuales para cada unidad de tiempo, o puede ser utilizado como un contador de propósito general. En el modo de calendario, el módulo RTC_A soporta varios registros para segundos, minutos, horas, días de la semana, días del mes, meses y años, en su formato BCD o hexadecimal según la configuración deseada. Para un debido funcionamiento, es importante que en este módulo estén correctamente configurados los registros RT0PS y RT1PS vinculados con la señal de reloj ACLK, misma que está configurada en frecuencia nominal de 32768 [Hz]. La salida del registro RTC_A queda definida como se presenta en la ecuación (19), la cual se compone de la ecuación (18) y en la cual se establece un segundo de intervalo final.

$$f(RTC_A) = \left(\frac{f(RT0PS_A)}{128} \right) [Hz] \quad (18)$$

$$f(RTC_A) = \left(\frac{f(ACLK)}{256 \times 128} \right) = 1[Hz] = 1[s] \quad (19)$$

En modo calendario, la correcta lectura de los registros de tiempo se realiza a partir de una interrupción generada por el propio módulo. Cuando la programación de sistema total involucra la intervención de otros módulos que de igual manera generan solicitudes de interrupción de sistema, resulta inconveniente la utilización del módulo calendario. Esto deja como alternativa la construcción vía software de una estructura semejante a los registros de tiempo mencionados utilizando al módulo RTC_A en modo contador. Al igual

que el modo calendario, el modo contador depende de la salida de los registros RT0PS y RT1PS respectivamente. Utilizando la misma configuración expuesta para el modo calendario, en este modo se puede usar también el resultado de la ecuación (19) para obtener un intervalo de un segundo.

Multiplicador de Hardware de 32-bits – 32-Bit Hardware Multiplier (MPY32)

MPY32 es un módulo periférico que permite realizar varias operaciones aritméticas con variables o registros de 8, 16 o 32 bits. Estas operaciones pueden ser: multiplicación con o sin signo, multiplicación con o sin acumulador y multiplicación con o sin números fraccionarios. Todo tipo de operación que utiliza el módulo MPY32, se encuentra definida por los registros operadores, los cuales definen el tipo de operación a realizar y el tamaño en bits de cada uno de los operadores. Dependiendo del tamaño de cada uno de los operadores, el resultado de dicha operación es almacenado en un registro hasta de 64 bits. Los principales operadores que definen la multiplicación a realizar se resumen en la *Tabla 4* en la que se muestra la descripción de cada uno de ellos.

Tabla 4: Módulo MPY32

Registros tipo OP1	Operación
MPY / MPYS	Multiplicación sin signo / Multiplicación con signo – 16 bits
MAC / MACS	Multiplicación con acumulador sin signo / Multiplicación con acumulador con signo – 16 bits
MPY32L/H / MPYS32L/H	Multiplicación sin signo / Multiplicación con signo – 32 bits
MAC32L/H / MACS32L/H	Multiplicación sin signo / Multiplicación con signo – 32 bits
Registros tipo OP2	Operación
OP2	Multiplicador de 16 bits
OP2L/H	Multiplicador de 32 bits

*Tomado de: SLAU259E (Texas Instruments, 2009b).

ADC12_A

ADC12_A es un módulo de alto rendimiento para la conversión análogo digital con una resolución de 12 bits en su conversión. Este módulo tiene una capacidad máxima de 200 [ksps] como velocidad de muestreo, un bloque de retención y muestreo [sample and hold] con períodos programables definidos por software o por la señal de salida del módulo TAX, canales individuales de conversión, modo de conversión simple o secuencial y almacenamiento del resultado de conversión en un registro de 16 bits. El diagrama de bloques de este módulo se presenta en el anexo B 1.1.

El funcionamiento y metodología de conversión para el módulo ADC12_A depende principalmente de los límites de voltaje preestablecidos. El resultado final de conversión para este módulo se lo puede expresar como se presenta en la ecuación (20):

$$N_{ADC12} = 4095 \times \left(\frac{V_{in} - V_{R+}}{V_{R+} - V_{R-}} \right) \quad (20)$$

Aquí N_{ADC12} es el resultado de la conversión, V_{in} es el valor de voltaje de entrada, V_{R+} es el voltaje de referencia positivo y V_{R-} es el voltaje de referencia negativo. Si no se activa ningún nivel externo de referencia, los valores por defecto son 3.3 [V] para la referencia positiva y 0 [V] para la referencia negativa.

Otro aspecto importante que caracteriza a este módulo es el tiempo de muestreo y conversión, ya que éste es iniciado por un cambio positivo en la señal SHI del módulo. Es importante mencionar que los ciclos necesarios de reloj se calculan en dependencia de la resolución de conversión del módulo para que haya una correcta conversión. Sin embargo, para que una conversión con resolución de 12 bits arroje un resultado correcto es necesario como mínimo 13 ciclos de reloj ADC12CLK. Una segunda señal existente en este módulo,

SAMPCON, es la responsable de activar y desactivar el tiempo de muestreo. Por esta razón, existen dos modalidades de muestreo: extendida y por pulso. En esta sección se detallará la modalidad por pulso representada en la *Figura 5*.

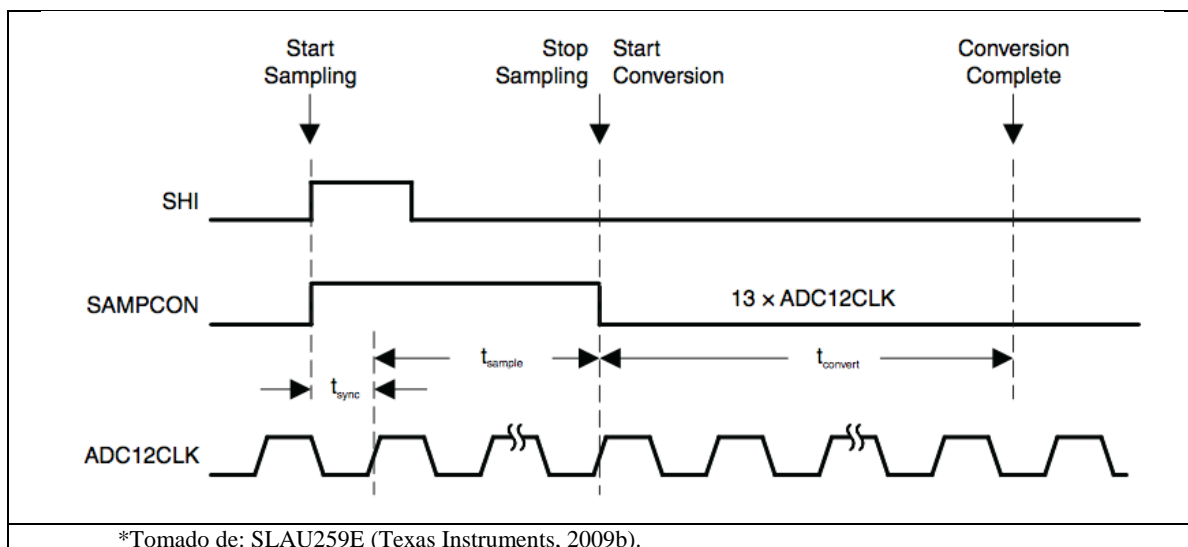


Figura 5: ADC12_A Pulse Mode

Como se puede observar en la figura, la señal SHI es la encargada de iniciar el temporizador de la conversión. Diferentes módulos de control son los encargados de definir el tiempo o intervalo de muestreo, es decir, la duración de la señal SAMPCON. El tiempo total de muestreo se lo puede definir como se presenta en la ecuación (42):

$$t_{\text{muestreo-total}} = t_{\text{sync}} + t_{\text{muestreo}} [s] \quad (21)$$

Aquí t_{sync} es el tiempo en el cual la señal de SAMPCON se sincroniza con la señal de reloj ADC12CLK y t_{muestreo} es el tiempo definido vía software a través de distintos registros de control. Posteriormente, una vez que se desactiva la señal SAMPCON, el registro de control ADC12SHTx establece un tiempo de conversión [múltiplos de ADC12CLK] para almacenar el resultado en los registros ADC12MEMx.

El módulo ADC12_A soporta también varias alternativas para la forma en la que se analizan cada uno de los canales de entrada. Este módulo puede analizar cada canal de forma individual con conversión simple, secuencial, repetitiva individual o repetitiva secuencial. El anexo B1.2 detalla el diagrama de funcionamiento de una de estas modalidades: la secuencial o autoscan. En esta modalidad, se escoge uno de los canales de conversión como el último miembro de la secuencia para indicar al conversor en cuál canal terminar el análisis. En los casos en que se trabaja con conversiones secuenciales o repetitivas, usualmente se configura el registro ADC12MSC para realizar múltiples conversiones en el sistema; esto siempre y cuando la conversión anterior haya finalizado y el módulo ADC12_A se encuentre activo.

Limitaciones eléctricas CC430F6137

Para evitar daños y malfuncionamiento en el microprocesador utilizado, es fundamental tomar en cuenta las siguientes limitaciones expuestas en la *Tabla 5* previo a realizar cualquier dimensionamiento adicional.

Tabla 5: Limitaciones Eléctricas

Descripción	Rango aceptable
Voltaje aplicado a DVCC y AVCC con referencia a Vss	-0.3 a 4.1 [V]
Voltaje aplicado a cualquier pin excluyendo VCORE, RF_O, RF_N y R_BIAS	-0.3 [V] a ($V_{CC} + 0.3$ [V]), 4.1 [V]
Voltaje aplicado a VCORE, RF_O, RF_N y R_BIAS	-0.3 [V] a 2.0 [V]
Niveles RF de entrada en los pines RF_P y RF_N	10 [dBm]
Corriente en el diodo de cualquier dispositivo terminal	± 2 [mA]
Temperatura de almacenamiento T_{stg}	$-55^{\circ}C$ a $150^{\circ}C$
Temperatura de juntura T_J	$95^{\circ}C$

*Tomado de: SLAU259E (Texas Instruments, 2009b).

Circuitos de medición y acondicionamiento

En la presente sección se detalla todo el procedimiento de diseño y dimensionamiento de los componentes que conformarán los circuitos sensores tanto para la señal de voltaje como la señal de corriente para cada uno de los cinco canales de medición. Para ello se toma en consideración principalmente las limitaciones eléctricas de la tarjeta de procesamiento presentadas en la *Tabla 5*. Finalmente, se realiza la simulación de cada sensor utilizando el programa ORCAD para poder así evaluar y comprobar su correcto funcionamiento.

Diseño, funcionamiento y simulación del sensor de voltaje

A continuación se propone un circuito para la medición del voltaje de cada carga. Para la elaboración del mismo se deben tomar en cuenta tanto las limitaciones eléctricas del microprocesador expuestas en la *Tabla 5*, así como la condición de mantener y conservar la onda de voltaje de entrada. Este circuito se muestra en la *Figura 6* y se compone de dos etapas.

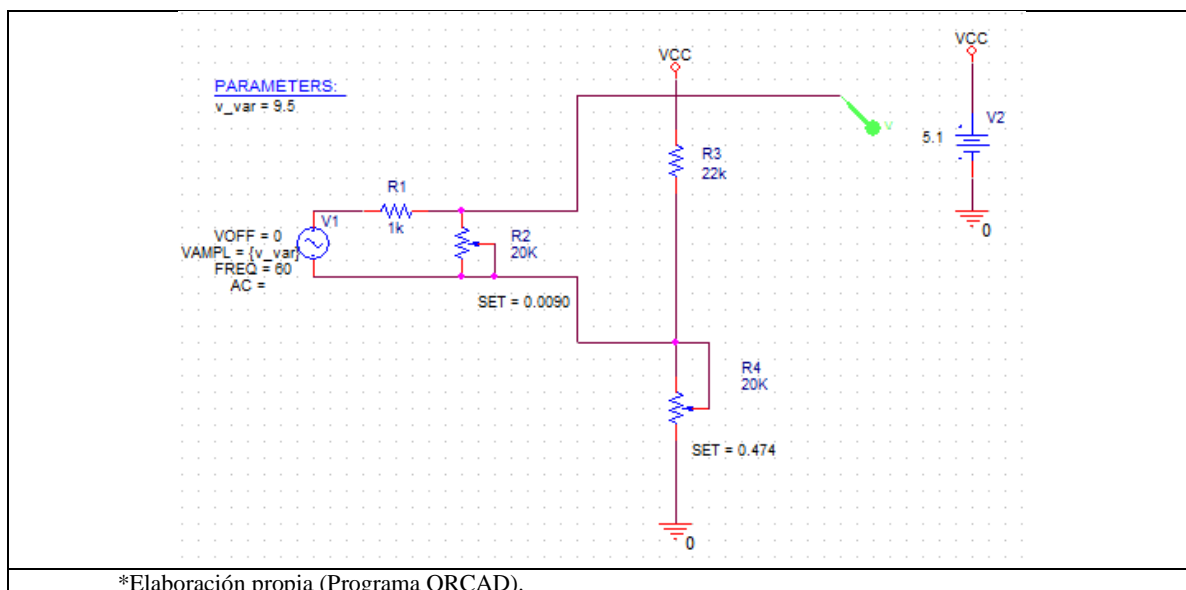


Figura 6: Circuito Sensor de Voltaje

La primera etapa consiste en la transformación o reducción de la onda de voltaje entrante. Como es de nuestro conocimiento, la empresa eléctrica distribuye para el usuario final una onda de voltaje que puede variar normalmente entre los 110 a 130 [V_{rms}]. Por esta razón, se utiliza un transformador TAMURA-3FS210 en configuración paralela para poder obtener así una onda alterna resultante con menor amplitud. Esta onda se encuentra simulada como la fuente alterna (V1) del esquemático presentado en la *Figura 6*. Posterior a realizar una prueba de cortocircuito al transformador, se requiere determinar el número de vueltas entre el devanado primario y el secundario, el cual en este circuito tiene un valor igual a 17. Esto quiere decir que el voltaje de salida del transformador se lo puede expresar como se muestra en las ecuaciones (22) y (23).

$$\frac{V_1}{V_2} = \frac{N_1}{N_2} = 17 \quad (22)$$

$$V_2 = \frac{V_1}{17} [V] \quad (23)$$

Aquí V_1 es el voltaje de devanado primario y V_2 es el voltaje del devanado secundario.

La segunda etapa se compone del circuito acondicionador de señal, el mismo que modifica la señal transformada reduciéndola a través de un puente divisor de tensión y añadiéndole un offset DC. Este último es el que impide que la onda tome valores negativos y la obliga a que se encuentre dentro del rango de entrada del microprocesador. Es importante recalcar que esta componente DC se la inserta con el uso de una fuente continua (V2) que en el circuito presente tiene un valor de 5.1 [V] y con un puente divisor compuesto por las resistencias R3 y R4.

Para dimensionar los componentes del primer puente divisor reductor compuesto por las resistencias R1 y R2, se debe considerar los siguientes parámetros de diseño:

- $V_{PIN_{max}} = \frac{(130 \times \sqrt{2})}{17} = 10.81 [V]$
- $V_{POUT_{max}} = 1.65 [V]$
- $R_1 = 1 [K\Omega]$

Si se considera al voltaje de salida como se expresa en las ecuaciones (24) y (25), se puede definir el valor de la resistencia R2 como se muestra en la ecuación (26).

$$V_{out} = \frac{R_2 \times V_{in}}{R_1 + R_2} [V] \quad (24)$$

$$V_{out} \times R_1 = R_2 (V_{in} - V_{out}) \quad (25)$$

$$R_2 = \frac{V_{out}}{V_{in} - V_{out}} \times R_1 [\Omega] \quad (26)$$

De igual manera, si se reemplaza los valores máximos de diseño en la ecuación (26) se obtiene un valor para la segunda resistencia de $R_2 = 180.13 [\Omega]$. Sin embargo, debido a la imprecisión de los materiales físicos esta resistencia se encuentra modelada en la *Figura 6* e implementada como un potenciómetro para compensar cualquier imprecisión.

Para dimensionar los componentes del segundo puente divisor se deben considerar los siguientes parámetros de diseño:

- $V_{IN} = 5.1 [V]$
- $V_{OUT} = 1.65 [V]$
- $R_3 = 22 [K\Omega]$

Siguiendo el mismo procedimiento utilizado para el primer puente divisor, se puede definir el valor de la resistencia R4 como se muestra en la ecuación (27).

$$R_4 = \frac{V_{out}}{V_{in} - V_{out}} \times R_3 [\Omega] \quad (27)$$

Si se reemplaza los valores máximos de diseño en la ecuación (27) se obtiene un valor para la cuarta resistencia de $R_4 = 10.52 [\text{K}\Omega]$. No obstante, esta resistencia también se encuentra modelada en la *Figura 6* e implementada como un potenciómetro para compensar cualquier imprecisión de los materiales físicos.

Para proceder con la simulación del circuito, es importante definir los valores SET pertenecientes a los potenciómetros de la *Figura 6*. El valor de resistencia para un potenciómetro utilizado en el programa ORCAD, se simula en dependencia del valor SET definido en las ecuaciones (28) y (29).

$$SET = 1 - \frac{R_{12}}{R_{tot}} \quad (28)$$

$$SET = \frac{R_{23}}{R_{tot}} \quad (29)$$

Aquí $R_{tot} = R_{12} + R_{23}$. Finalmente, utilizando este resultado se puede deducir lo siguiente:

- $SET_{R_2} = \frac{R_2}{20000} = 0.0090$
- $SET_{R_4} = 1 - \frac{R_4}{20000} = 0.474$

La simulación del circuito se realiza para las siguientes tres distintas amplitudes de entrada:

- $V_{amp1} = 9.15[V]$ equivalente a una onda de entrada de 110 $[V_{rms}]$

- $V_{amp2} = 9.98[V]$ equivalente a una onda de entrada de 120 [V_{rms}]
- $V_{amp3} = 10.81[V]$ equivalente a una onda de entrada de 130 [V_{rms}]

El resultado obtenido se expone en el anexo C1.1. Como se puede observar en primer lugar, la onda resultante para cada uno de los casos se encuentra dentro de los límites eléctricos del microprocesador, lo que asegura su correcto funcionamiento y conversión. En segundo lugar, se puede observar que la amplitud de la onda de salida es proporcional a la amplitud de la onda de entrada. Por último, se observa que para todos los casos se tiene una componente DC cercana a su valor teórico de 1.65 [V]. Con esta simulación se puede concluir el correcto funcionamiento del sensor así como el correcto dimensionamiento de todas sus partes.

Diseño, funcionamiento y simulación del sensor de corriente

El análisis de la onda de corriente de entrada presenta mayor dificultad que la onda de voltaje debido a que la onda de corriente depende del tipo de carga utilizada y puede sufrir varias alteraciones o distorsiones en su forma. Por esta razón, el núcleo del sensor de corriente propuesto se basa en la utilización del ACS712-20A-T ya que es un sensor de corriente tipo efecto hall capaz de medir corrientes alternas o continuas de hasta 20 [A]. El diagrama funcional de bloques de este sensor se presenta en el anexo C1.2. El funcionamiento del mismo, se lo puede explicar a través de la gráfica presentada en la *Figura 7*.

A partir de esta gráfica se pueden obtener dos conclusiones fundamentales sobre el comportamiento del sensor. En primer lugar, se puede notar que cuando no se ha detectado la presencia de corriente, el voltaje de salida es igual a la mitad del voltaje de alimentación

(V_{CC}). En segundo lugar, se puede observar un comportamiento lineal de salida, lo cual facilita la correcta medición de corriente y evita el uso de factores por no linealidad. De igual forma, a partir de esta gráfica se puede concluir que la sensibilidad del sensor es de 100 [mV] por cada 1 [A] de detección. Es decir, a escala completa de medición, la mínima salida del sensor es aproximadamente igual a 0.5 [V] y la máxima salida del sensor es aproximadamente igual a 4.5 [V].

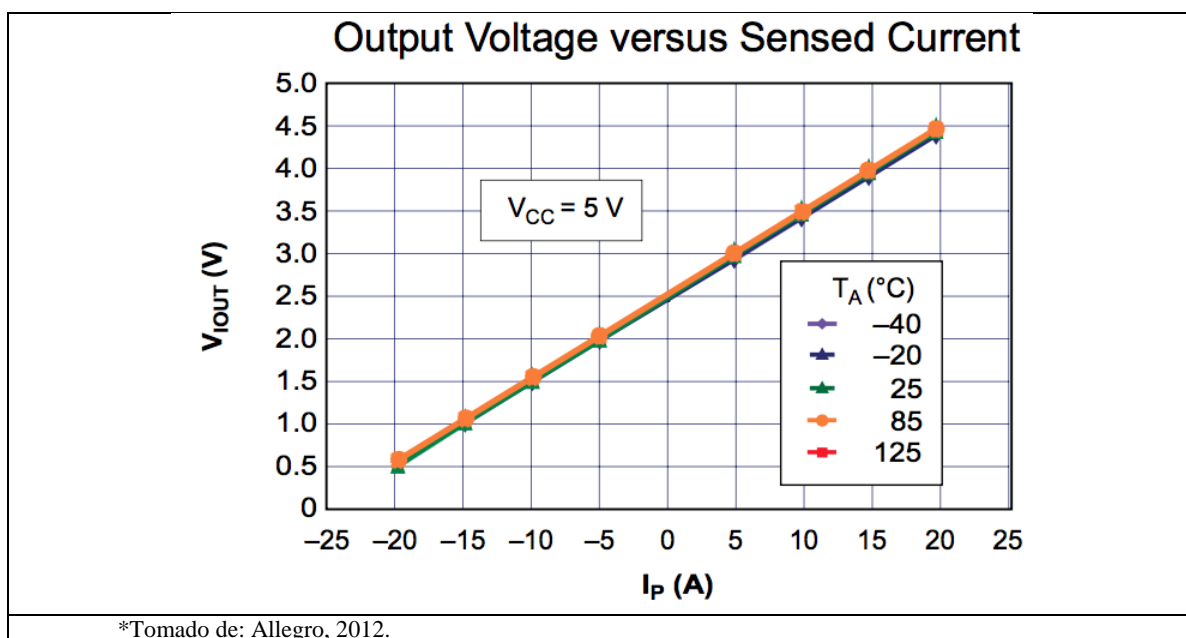


Figura 7: Comportamiento ACS712-20A-T

Debido a que el rango establecido de salida del sensor no cumple con las restricciones eléctricas del microprocesador utilizado, se procede a realizar una segunda etapa de acondicionamiento de señal para que un nuevo rango de salida pueda ser utilizado en el puerto de entrada del procesador. El circuito de acondicionamiento propuesto se presenta en la Figura 8. En este circuito la fuente de voltaje alterna (V1) simula la salida del sensor ACS712. De igual manera, la fuente de voltaje constante (V2) simula la misma fuente utilizada y mencionada en el sensor de voltaje.

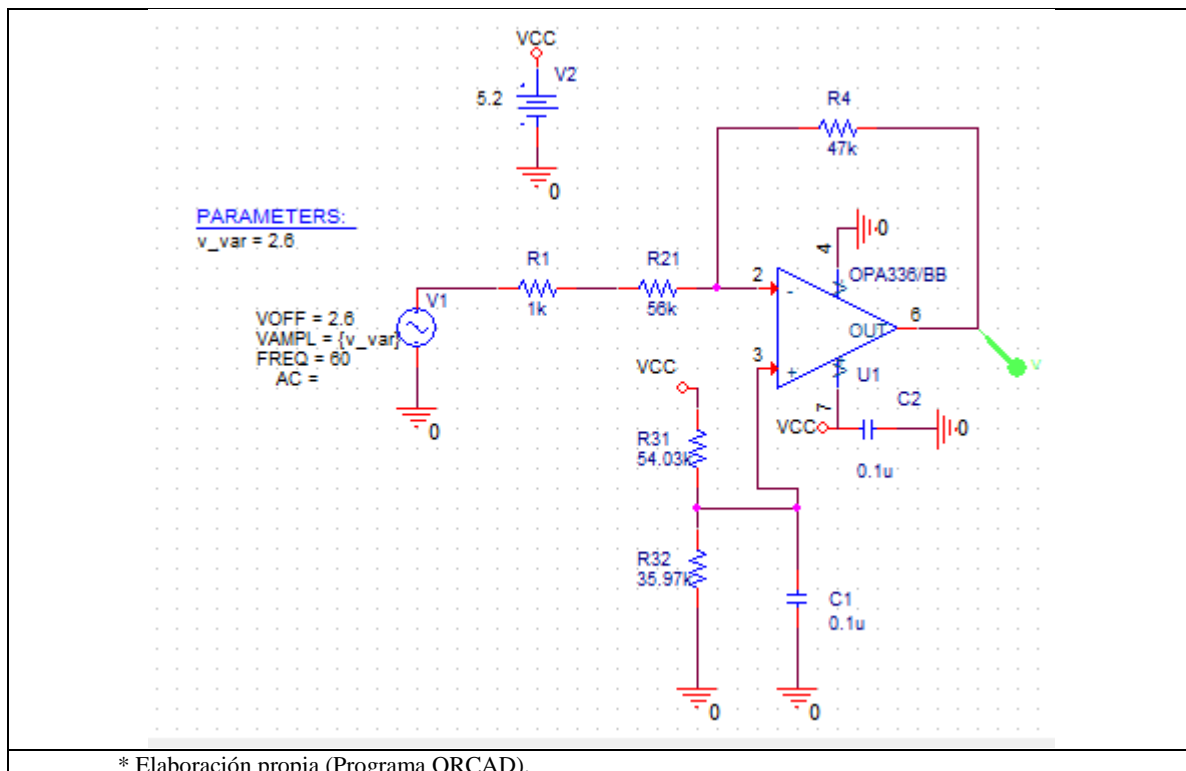


Figura 8: Sensor de Corriente

A continuación, se detalla las ecuaciones de diseño para el dimensionamiento de las resistencias utilizadas: R1, R21, R4, R31 y R32. En este caso se asumen condiciones ideales para el amplificador operacional de la serie OPA33x utilizado. Si se analiza detenidamente el circuito, se puede notar que el voltaje de la entrada no inversora al OPA33x se la define como se muestra en la ecuación (30).

$$V_{inv} = \frac{V_{cc}R_{32}}{R_{31} + R_{32}} [V] \quad (30)$$

Utilizando este resultado, se define el voltaje de salida como se deduce de las ecuaciones (31), (32) y (33):

$$I_{inv} = I_{R1} + I_{R4} = \left(\frac{V_1 - V_{inv}}{R_1 + R_{21}} \right) + \left(\frac{V_{out} - V_{inv}}{R_4} \right) = 0 \quad (31)$$

$$\frac{R_4(V_1 - V_{inv}) + (R_1 + R_{21})(V_{out} - V_{inv})}{(R_1 + R_{21})(R_4)} = 0 \quad (32)$$

$$V_{out}(R_1 + R_{21}) = V_{inv}(R_1 + R_{21} + R_4) - R_4V_1 \quad (33)$$

Sin embargo, debido a que $V_{inv} = V_{ninv}$ en este caso se tiene la ecuación (34).

$$V_{out} = \frac{1}{(R_1 + R_{21})} [V_{ninv}(R_1 + R_{21} + R_4) - R_4V_1] [V] \quad (34)$$

Considerando el resultado anterior, las limitaciones eléctricas del microprocesador, el comportamiento lineal del sensor ACS712 y estableciendo los mismos parámetros de comportamiento utilizados en el sensor de voltaje, se puede, tras un análisis matemático, establecer una función que relacione el voltaje de entrada con el de salida y proporcione el comportamiento deseado. Dicha función se presenta en la ecuación (35).

$$V_{out} = -0.825V_{in} + 3.795 \quad (35)$$

Así mismo, utilizando el resultado expuesto en la ecuación (34), la función de la ecuación (35) y un programa de MATLAB basado en la convergencia numérica, se pueden establecer los valores de R_{21} y V_{ninv} que más se ajustan a la función utilizada. Este programa se encuentra en el Anexo C1.3. Los resultados que arroja el programa son:

- $R_{21} = 55.9 [K\Omega]$
- $V_{ninv} = 2.0785 [V]$
- $R_1 = 1 [K\Omega]$ (Parámetro de diseño)
- $R_4 = 47 [K\Omega]$ (Parámetro de diseño)

Al utilizar este último resultado expuesto, la definición de la ecuación (30) y considerando la siguiente relación: $R_{31} + R_{32} = 90[K\Omega]$, se pueden obtener los valores

para las resistencias R_{31} y R_{32} como se deduce de las ecuaciones (36) y (37) y se muestra en las ecuaciones (39) y (38) respectivamente.

$$V_{ninv} = \frac{V_{cc}R_{32}}{R_{31} + R_{32}} = 2.0785[V] \quad (36)$$

$$\frac{R_{32}(5.2 - 2.0785)}{2.0785} = R_{31} \quad (37)$$

$$R_{32} = 35.97 [K\Omega] \quad (38)$$

$$R_{31} = 54.03 [K\Omega] \quad (39)$$

Es importante mencionar que las resistencias R_{21} , R_{31} y R_{32} son implementadas como las resistencias internas de un potenciómetro. Para el caso de R_{21} , esta resistencia representa la total ya que R_{12} se encuentra en cortocircuito.

La simulación del circuito se realiza, al igual que con el circuito de voltaje, para tres distintas amplitudes de entrada. Estas son:

- $V_{amp1} = 0[V]$ en caso de ausencia de corriente detectada
- $V_{amp2} = 1[V]$ equivalente a una onda de entrada de 10 [A]
- $V_{amp3} = 2[V]$ equivalente a una onda de entrada de 20 [A]

El resultado obtenido se expone en el anexo C1.4. Primero, se puede observar que la onda resultante para cada uno de los casos se encuentra dentro de los límites eléctricos del microprocesador, lo cual asegura su correcto funcionamiento y conversión. Segundo, se puede observar que la amplitud de la onda de salida es proporcional a la amplitud de la onda de entrada. Posteriormente, se observa también que para todos los casos se tiene una componente DC cercana a su valor teórico de 1.65 [V].

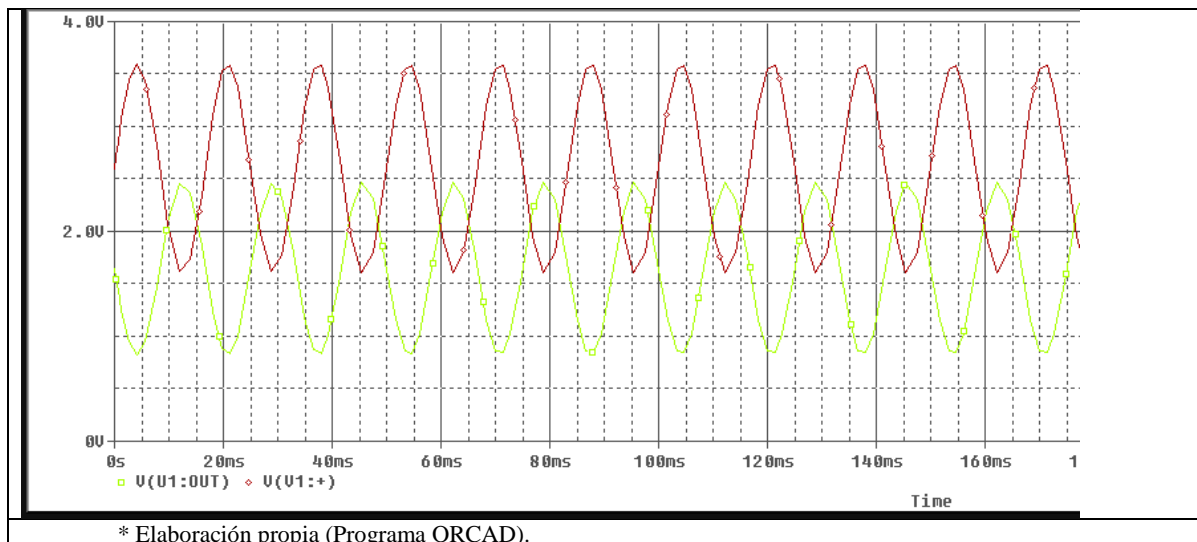


Figura 9: Comportamiento Sensor Corriente

Con esta simulación se puede observar que la onda de salida para este sensor se comporta de manera similar que la onda resultante del sensor de voltaje. Sin embargo, debido a la pendiente negativa del circuito acondicionador ocasionada por la entrada de la señal en el puerto inversor del OPA33x, la fase de la onda de salida [señal verde] se invierte con respecto a la fase de la onda de entrada [señal roja]. Esta inversión se muestra en la *Figura 9*. Este inconveniente puede ser solucionado en la práctica si se polariza al sensor ACS712 de manera inversa, con lo cual se elimina la inversión de fase en la salida del circuito acondicionador.

Diseño y planificación de la tarjeta PCB (Printed Circuit Board)

Una vez concluido el diseño de cada sensor y comprobado su correcto funcionamiento, se continúa con el diseño de una tarjeta impresa [PCB] que contenga a todos los circuitos antes mencionados. El diseño de esta tarjeta se realiza utilizando el programa Altium Designer, el cual permite al usuario realizar un esquemático del circuito y plasmar este esquemático dentro de una tarjeta impresa, similar a lo que hace el programa ORCAD. Para la elaboración de esta tarjeta se considera el espacio necesario

para cada componente, las normativas eléctricas para evitar fallas o cortocircuitos, y se respeta el espacio físico real disponible para la PCB diseñada. El programa escogido permite diseñar la distribución espacial de cada uno de los elementos dentro de la tarjeta, todos los caminos de conducción, las áreas de no conductividad y la estructura física de la tarjeta como tal.

El esquemático final de todo el sistema de medición propuesto se presenta en el anexo D1.1. En esta figura se puede observar la implementación de un sensor de voltaje y cinco sensores de corriente, tal como ya se ha discutido. Además de todos estos circuitos, se pueden observar elementos adicionales que se encuentran relacionados con la alimentación del circuito, con la protección del circuito y con diferentes puertos tanto de entrada como de salida. Estos puertos se adaptan con otras estructuras del sistema propuesto como se discutirá en la siguiente sección. Posterior al esquemático final, se diseña la tarjeta PCB como se muestra en la *Figura 10* y *Figura 11*. En estas gráficas, se pueden observar con detalle los principales componentes del sistema de medición implementado.

Por un lado, en la parte derecha de la *Figura 10* se puede observar el sistema adicional de alimentación y protección compuesto por las siguientes características:

- Puerto de entrada para línea, neutro y tierra [Puerto: Pin_AC_DC].
- Espacio dedicado a un porta-fusible conectado a línea.
- Espacio dedicado a un transformador AC-DC [RECOM RAC02-05SC], el cual se alimenta con línea y neutro y tiene un voltaje DC teórico de salida de 5 [V].

- Espacio dedicado a un conversor DC_DC [módulo independiente], el cual es alimentado por los 5 [V] del sistema y tiene un voltaje de salida de 3.3 [V] que es utilizado para alimentar la tarjeta EM430F6137RF900.

Por otro lado, en la parte superior izquierda de la *Figura 10* se puede observar la distribución espacial de los cinco sensores de corriente, y en la parte inferior izquierda la del sensor de voltaje. Por último, este PCB se ha diseñado también pensando en la estructura física de la tarjeta EM430F6137RF900. Esta es la razón por la cual se justifica la presencia de los puertos Port2 y COM9 en el lado derecho de la *Figura 10* para interconectar estas dos tarjetas.

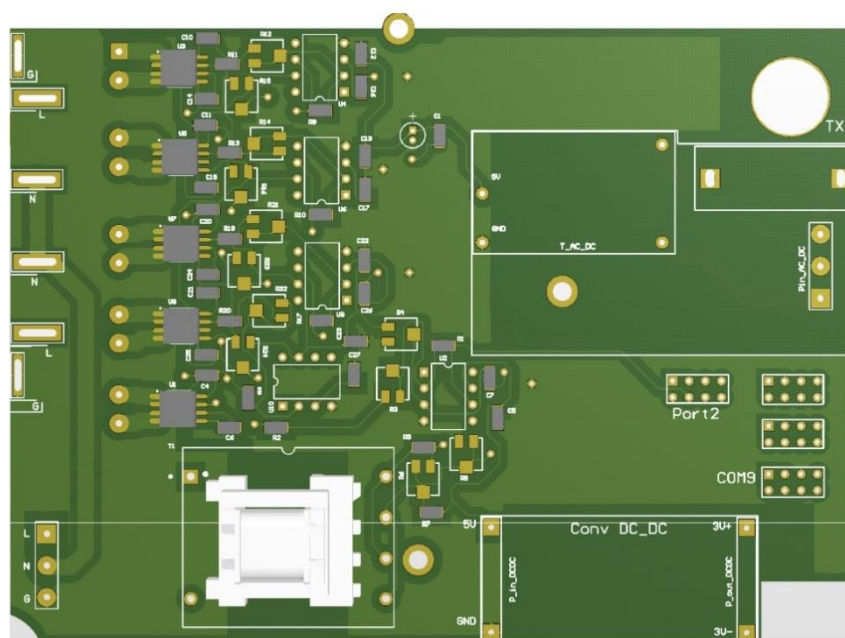


Figura 10: PCB (vista superior)

Cabe recalcar que dentro del diseño de la tarjeta PCB se tomó en consideración ciertas normativas para prevenir un malfuncionamiento o para optimizar la estabilidad de la tarjeta. Dentro de las características de prevención se implementó diferentes grosores y espacios para los caminos de alta corriente de manera que se diferencien de los caminos de

baja corriente. En los caminos de alta corriente se consideró un espacio o grosor de línea mayor o igual a 2.5 [mm]; mientras que en los caminos de baja corriente se consideró un espacio o grosor de línea mayor o igual a 0.25 [mm]. Dentro de las características de optimización por su parte, se eliminó todo ángulo recto para los caminos y se implementó dos polígonos de tierra. Estos últimos utilizan todo el espacio no utilizado del PCB como tierra y separan la tierra del circuito de potencia con la tierra del circuito digital. Todo lo explicado se puede observar con mayor claridad en la *Figura 11*.

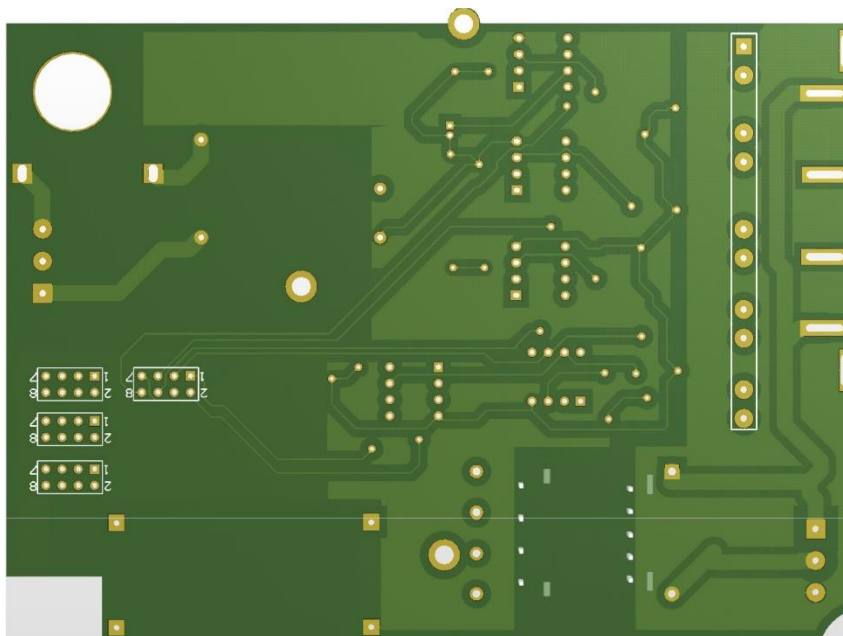


Figura 11: PCB (vista inferior)

Modelamiento tridimensional del sistema de adquisición de datos

Para concluir con el proceso de la implementación de hardware del sistema de medición diseñado se realiza en el programa Solidworks un modelo tridimensional de toda la estructura que envolverá a la tarjeta PCB, EM430F6137RF900 y los diferentes puertos donde se conectarán las cargas estudiadas. El programa mencionado permite importar el modelo 3D de la tarjeta PCB diseñada en el programa Altium Designer y unirlo a otras

estructuras diseñadas. Para este sistema, se ha seleccionado una estructura de tipo regleta ya que de esta manera se puede establecer cada tomacorriente como un puerto de conexión hacia la carga y también como uno de los cinco puertos de medición para los sensores de corriente. En la *Figura 12* se pueden observar estos cinco tomacorrientes, además de un interruptor general de encendido o apagado del sistema y un amplio espacio superior en donde se colocarían las tarjetas de adquisición y procesamiento de datos.

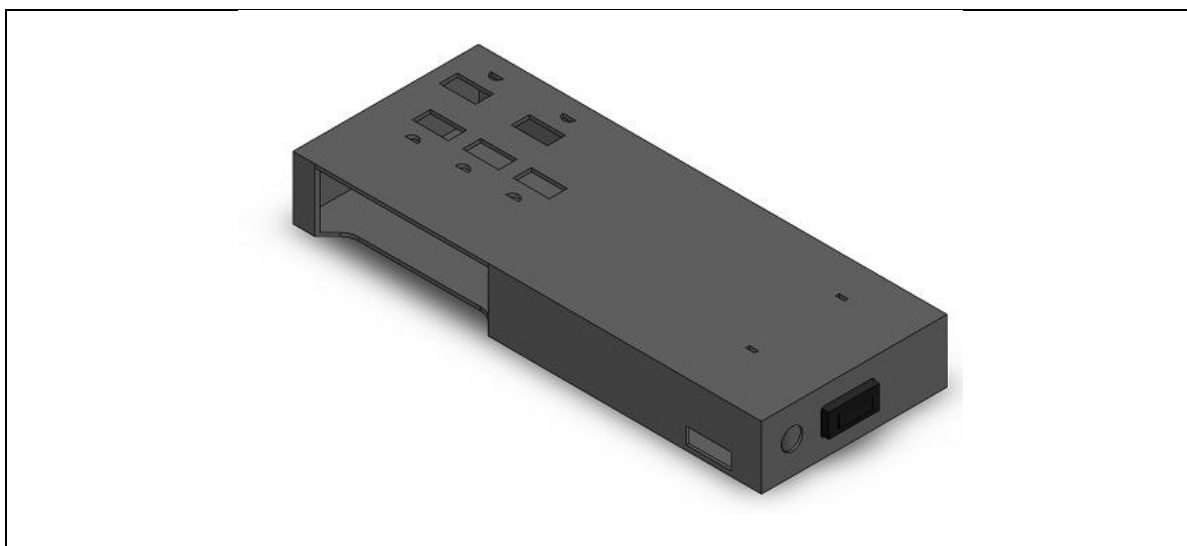


Figura 12: Estructura tipo Regleta

Si se consideran todas las características y limitaciones espaciales propuestas en la estructura diseñada de tipo regleta, se modifica cada una de las tarjetas implementadas en las secciones anteriores y se las distribuye en el interior de la estructura expuesta en la *Figura 12*, se tiene como resultado el sistema de medición presentado en la *Figura 13*. Con estas imágenes se puede concluir con la implementación del hardware. Ésta tiene como resultado una estructura compacta y eficiente que permite al usuario programar o acceder a la tarjeta EM430F6137RF900 de forma fácil y segura.

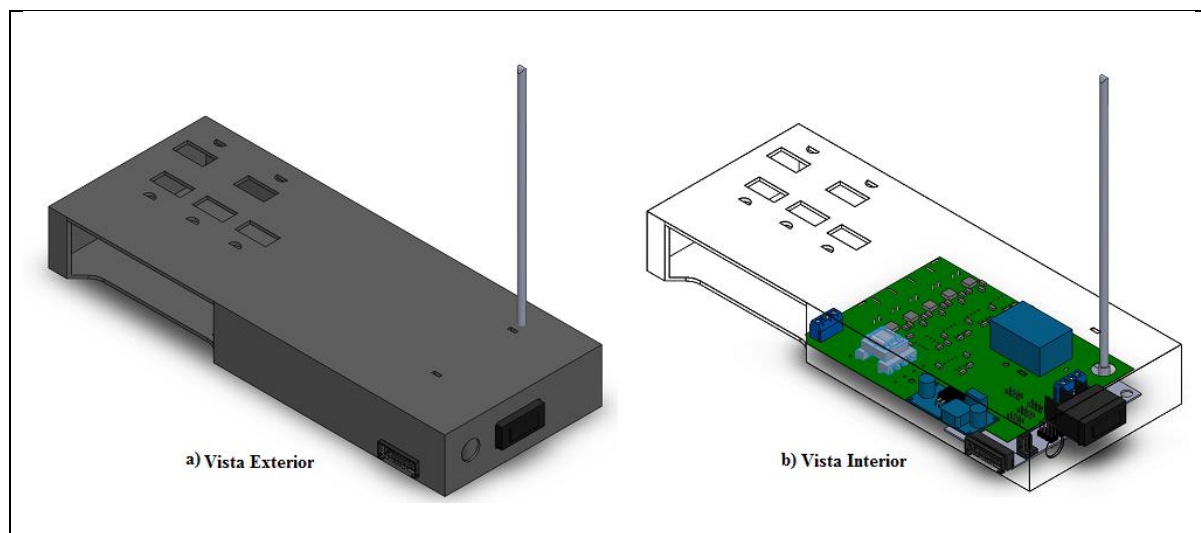


Figura 13: Sistema de medición completo

Capítulo 4:

Programación del DAQ

Implementación de Software

El presente capítulo resume los algoritmos diseñados e implementados en programación de alto y bajo nivel aplicados a la tarjeta EM430F6137RF900. Para esta programación se utiliza el programa Code Composer Studio conectado a un MSP430-USB-Debug-Interface. Todos estos algoritmos tienen como objetivo la correcta transformación análoga digital de señales y su posterior procesamiento. Lo que se desea al final es lograr calcular de forma aproximada la potencia que consume cada carga conectada al sistema de medición, junto con otras variables relacionadas.

Programación y procesamiento digital de las señales

El desarrollo de todos los algoritmos que se presentan a continuación tiene como sustento teórico dos trabajos previos realizados por Texas Instruments. El primero, “Energy Meter Code Library for 1-Phase to 3-Phase Using MSP430 Family” (Mesganaw, 2012), es una nota técnica en la que se pone a disposición del público en general todos los códigos desarrollados en el trabajo. Esta librería de códigos contiene todas las funciones, escritas en C o Assembler que utiliza el autor tanto para el manejo y adquisición de datos, así como para el procesamiento matemático de los mismos. Debido a que el microprocesador utilizado para el proyecto de la presente tesis [CC430F6137] mantiene la misma estructura global de programación y el mismo manejo de registros utilizados en la librería, se puede utilizar gran parte del código registrado en ésta para la programación del proyecto. No obstante, como la nota técnica en mención no fue diseñada para este tipo de

procesador es necesario realizar un proceso de adaptación del código para que funcione con el CC430F6137, y esto es lo que se realiza en el desarrollo de las siguientes secciones.

Uno de los mayores inconvenientes que la librería mencionada presenta para este proyecto es que fue diseñada utilizando conversores análogo digital de tipo sigma-delta con una resolución de 16 o 24 bits. En el microprocesador CC430F6137 el conversor análogo digital no es de tipo sigma-delta y que éste contiene una resolución máxima de 12 bits. Es por esta razón que para la digitalización de los datos se utiliza un segundo trabajo: “Three-Phase Electronic Watt-Hour Meter Design Using MSP430” (Underwood, Frangline & Chan, 2008). Éste contiene, en su gran mayoría, las mismas funciones descritas en el primer trabajo pero aporta nuevas funciones basadas en un conversor de 12 bits que es compatible con el procesador CC430F6137 utilizado. En conclusión, los algoritmos presentados en las siguientes secciones son el resultado de una combinación entre los dos trabajos previos realizados por Texas Instruments y un aporte propio de funciones que se adaptan tanto al microprocesador utilizado como al sistema de medición propuesto.

Rutinas y algoritmos

En esta presente sección se discute sobre las rutinas y algoritmos para procesar y calcular las variables de salida características de un medidor electrónico monofásico con cinco canales de entrada. Este medidor utiliza el dispositivo CC430F6137 como ya se mencionó y como se muestra en el diagrama de bloques de la *Figura 14*. Es importante mencionar que la conexión de la carga analizada con el sensor de voltaje se realiza en paralelo y con los sensores de corriente se las realiza en serie.

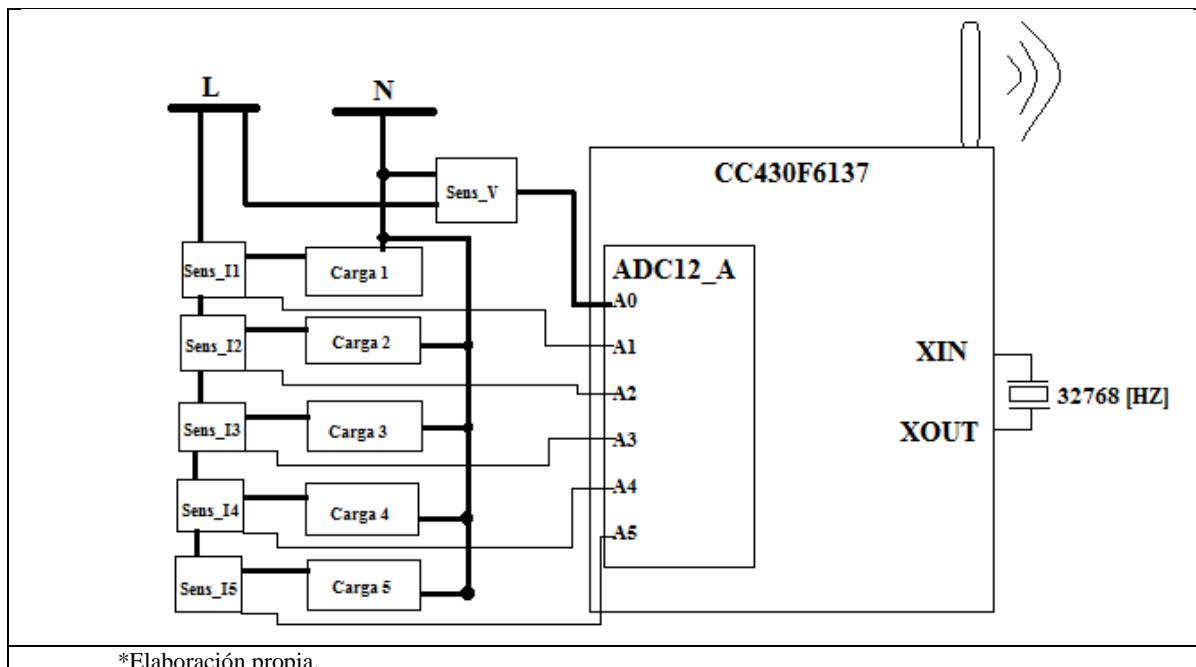


Figura 14: Diagrama Bloques Sistema Medición

Las distintas rutinas o algoritmos implementados se encuentran concentrados principalmente en tres rutinas de mayor importancia: main, background y foreground. El archivo background.c es de suma importancia para el correcto funcionamiento del sistema de medición ya que contiene todos los parámetros de tiempo y sincronización. Las funciones presentes en background.c utilizan una subrutina de interrupción del temporizador TA0 para inicializar la adquisición de muestras y transformación analógica digital de los datos de entrada para cada uno de los cinco canales de corriente y para el canal de voltaje. Estas muestras son posteriormente acumuladas y procesadas en un sistema de estructuras de variables separadas por cada canal de medición que se divide en dos grupos principales. El primer grupo, la estructura no volátil, almacena variables y parámetros constantes de sistema y se identifica como 'phase_nv'. El segundo grupo, la estructura volátil, contiene todas las variables de almacenamiento continuo de las mediciones realizadas por período de muestreo y se identifica como 'phase'.

Una vez acumuladas suficientes muestras por cada canal de medición, las funciones presentes en foreground.c son utilizadas para calcular los valores de potencia activa y aparente consumida, la frecuencia de la onda de voltaje de entrada y los valores efectivos tanto para voltaje como para corriente. Por motivos de organización y estructura, existen también otros archivos adicionales, los cuales configuran elementos importantes del sistema tales como el reloj a tiempo real RTC [emeter-rtc.c] y la inicialización de variables y registros [emeter-setup.c]. Además, todas las funciones matemáticas utilizadas se encuentran en archivos por separado. Finalmente, después de cada inicialización del sistema, cuando éste no se encuentra dentro de ninguna subrutina de interrupción, el programa entra en el lazo principal, main.c, donde se espera que la interrupción del temporizador recolecte los datos. Este último código es el encargado de activar o desactivar la ejecución de foreground.c.

Ficheros o archivos de cabecera (header file).h

Los archivos de cabecera contienen la declaración directa de las variables, subrutinas, estructuras y funciones utilizadas en los diferentes códigos de ejecución. Un resumen de cada uno de estos ficheros utilizados se presenta en la *Tabla 6*.

Tabla 6: Archivos de cabecera

Archivo de cabecera	Descripción
msp430.h	<ul style="list-style-type: none"> • Incluye el archivo cc430f6137.h el cual contiene la declaración de toda la arquitectura del microcontrolador utilizado
metrology-parms.h	<ul style="list-style-type: none"> • Definición de todos los módulos de medición activos e inactivos • Definición de parámetros propios de CC430F6137 • Definición de parámetros de calibración

emeter-toolkit.h	<ul style="list-style-type: none"> • Declaración directa de todas las funciones matemáticas utilizadas
emeter-metrology-structs.h	<ul style="list-style-type: none"> • Definición de parámetros característicos para la conversión análoga digital • Declaración de variables para almacenamiento de voltaje, corriente y potencia • Declaración directa de las estructuras de almacenamiento de datos por canal • Codificación numérica para los diferentes estados del sistema de medición • Declaración indirecta de las distintas funciones utilizadas por foreground.c
emeter-metrology.h	<ul style="list-style-type: none"> • Declaración indirecta de subrutinas de aplicación
emeter-rtc.h	<ul style="list-style-type: none"> • Declaración directa de la estructura RTC del sistema • Declaración indirecta de las funciones asociadas con el reloj a tiempo real

Inicialización del sistema (emeter-setup.c)

En esta sección se explica la subrutina que se encarga de inicializar el programa, es decir la que realiza todas las configuraciones iniciales de los registros y asigna los valores iniciales de las variables de mayor relevancia dentro del código. El algoritmo de inicialización del sistema implementado se resume en un diagrama de bloques que se presenta a continuación en la *Figura 15*. El proceso de mayor relevancia dentro de este algoritmo de inicialización es, sin duda alguna, la configuración de los diferentes registros de sistema que caracterizan y describen el funcionamiento del microprocesador. Por esta razón, se detalla la configuración de cada uno de ellos en las siguientes secciones.

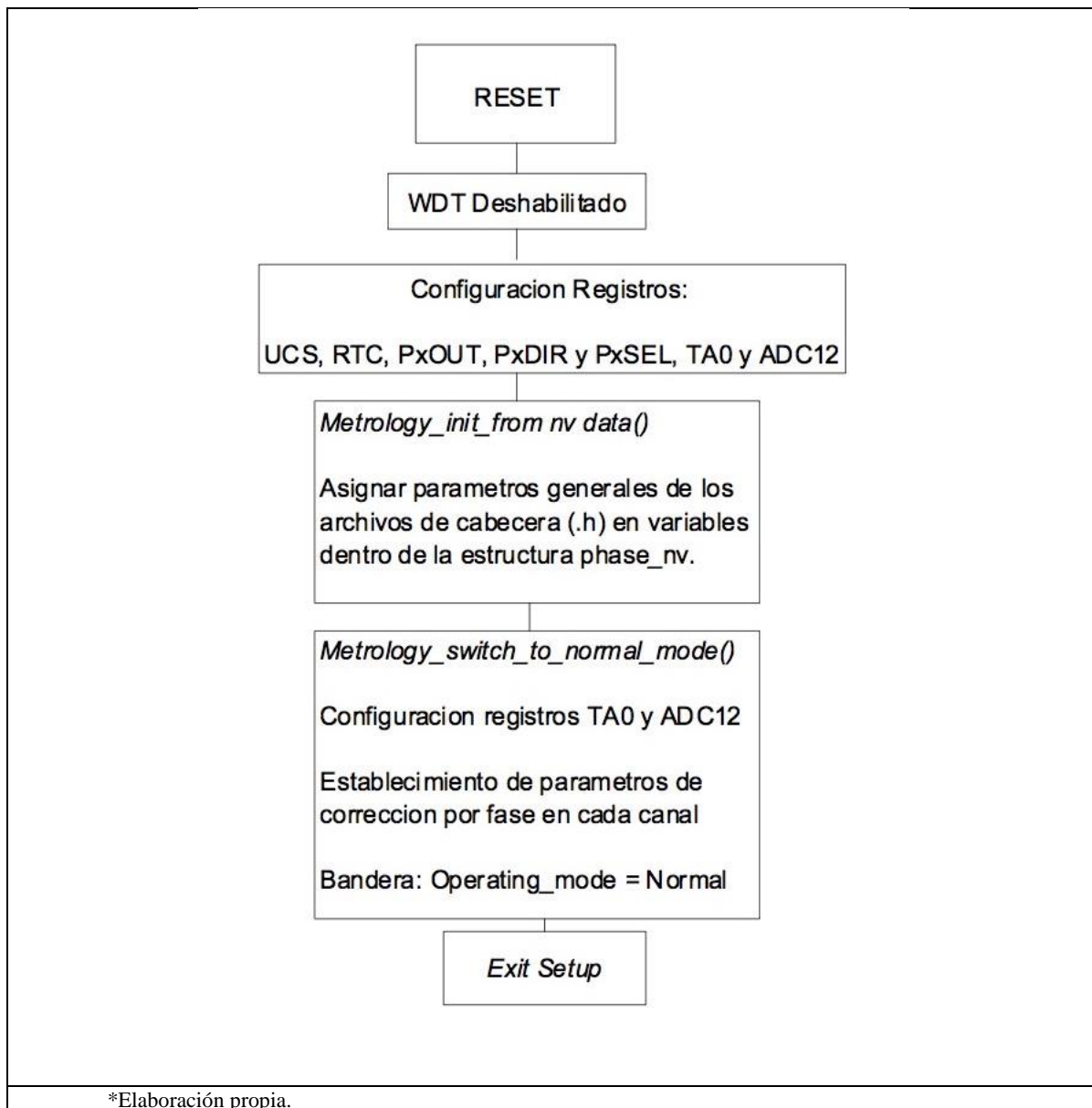


Figura 15: Flujograma Inicialización (setup)

Sistema de Reloj Unificado (UCS)

En base a lo mencionado en la sección anterior y lo expuesto en la *Tabla 2*, el módulo UCS se configura para que la señal de reloj ACKL funcione a partir de la fuente XT1CLK y para que las señales SMCLK y MCLK funcionen a partir de la fuente DCOCLK. Esta última fuente mencionada se la configura para que funcione a una frecuencia nominal de 32 [Mhz].

Reloj a tiempo real (RTC_A)

Al módulo RTC_A se lo configura en modo contador y con una salida en cascada como se presenta en la ecuación (19), la cual da como resultado una frecuencia de interrupción de un segundo. Esta configuración es de suma importancia ya que a través de la interrupción generada por este módulo se actualiza en cada segundo la estructura encargada de almacenar el tiempo del sistema.

ADC12

Teniendo en cuenta la descripción realizada en secciones anteriores sobre el módulo ADC12, se configuran los registros de modo que se obtenga las siguientes características:

- La conversión es múltiple con formato secuencial simple.
- La conversión es activada con la señal de salida del módulo TAX.
- La conversión es almacenada desde el registro ADC12MEM0 en adelante.
- Cada conversión toma 16 ciclos del reloj ADC12CLK.
- La referencia positiva de conversión es AVCC (3.3 [V]) mientras que la negativa es GND (0 [V]).
- Los canales de conversión se asignan como se muestra en el diagrama de la *Figura 14*.

TA0

La configuración del módulo TA0 junto con la del módulo de conversión ADC12, representan el núcleo del funcionamiento del sistema de medición ya que existe una dependencia entre un módulo y el otro. Muchos de los sistemas de medición contienen una forma manual [vía software] de activar o desactivar la conversión análoga digital. Sin

embargo, esta metodología en muchas ocasiones inserta problemas de ruido y desincronización en la medición. Por esta razón, se ha programado una forma automática [vía hardware] de activar o desactivar la conversión análoga digital al establecer que la salida del módulo TA0 sea la señal de control del módulo ADC12.

Teniendo en cuenta la descripción realizada anteriormente sobre este módulo, se configuran los registros de modo que se obtenga las siguientes características:

- Configuración del temporizador como contador ascendente [Up mode] como se presenta en la *Figura 3*.
- Fuente de señal de reloj como ACLK.
- Configuración del registro TA0CCR0 igual a 9 y TA0CCR1 igual a 7, lo cual da como resultado una frecuencia de muestreo de $3276.8 \left[\frac{\text{samples}}{\text{s}} \right]$.
- Salida tipo set/reset como se presenta en la *Figura 4*.

Para entender a detalle el motivo de las constantes utilizadas para los registros TA0CCR0/1 y su relación con la frecuencia de muestreo, se presenta un ejemplo con cuatro canales de conversión como se muestra en el diagrama de la *Figura 16*.

En la gráfica se puede observar que el contador TAR inicia desde un valor igual a cero hasta llegar a su valor máximo igual al valor del registro TACCR0. Este último define el tiempo de activación del módulo ADC12 y, por ende, la frecuencia de muestreo. La ecuación (40) que se presenta a continuación muestra justamente como se calcula esta frecuencia.

$$f_{\text{muestreo}} = \frac{f_{\text{ACLK}}}{\text{TAR_max} + 1} = \frac{32768}{10} = 3276.8 \left[\frac{\text{samples}}{\text{s}} \right] \quad (40)$$

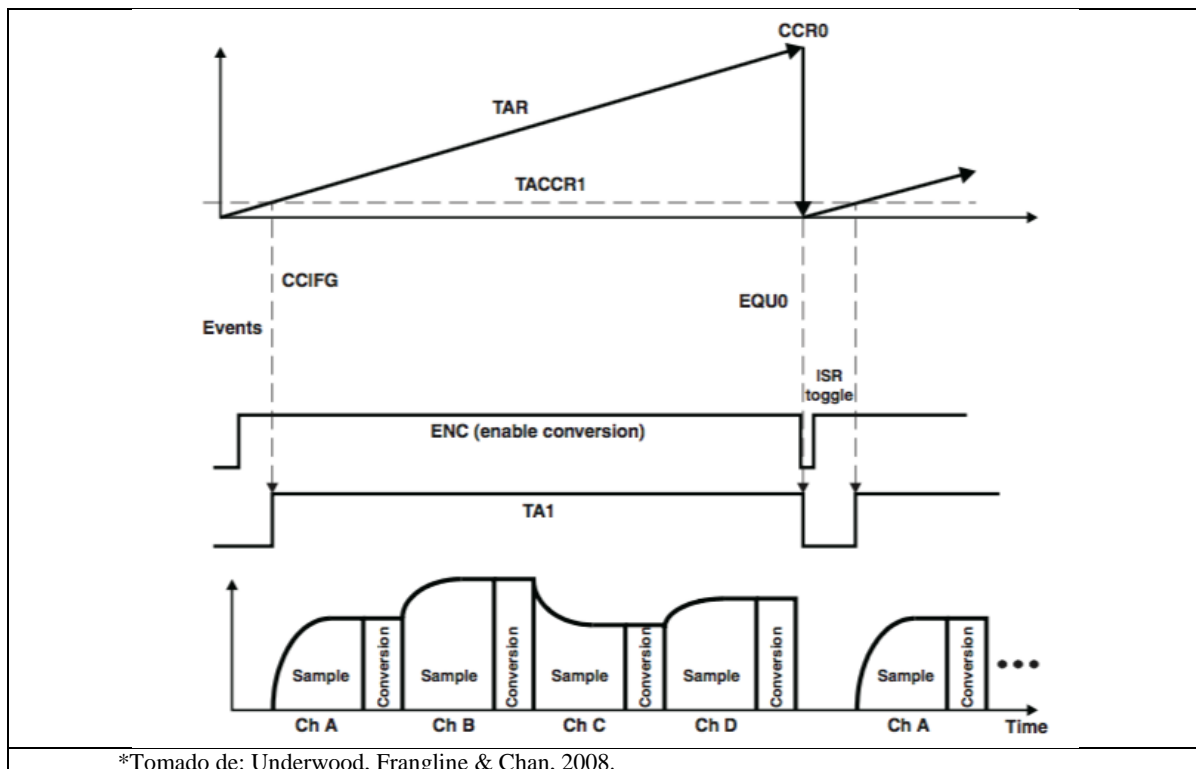


Figura 16: Proceso de conversión y muestreo

Así mismo se puede observar que el valor establecido en el registro TA0CCR1 limita la duración de la señal que define la salida del módulo TAX y, por ende, el tiempo de muestreo y conversión de datos para cada uno de los seis canales analizados. Se debe recordar que en el momento en el que el contador TAR llega a su valor máximo, la bandera de interrupción se activa y la subrutina de interrupción se ejecuta. Sin embargo, para cuando la subrutina de interrupción sucede, un completo set de datos se encuentra listo para ser procesado.

Proceso de background del sistema (background.c)

El algoritmo background.c del sistema implementado se resume en el diagrama de bloques presentado en la *Figura 17*. Posterior a este diagrama se detallan en las siguientes secciones los procesos de mayor relevancia.

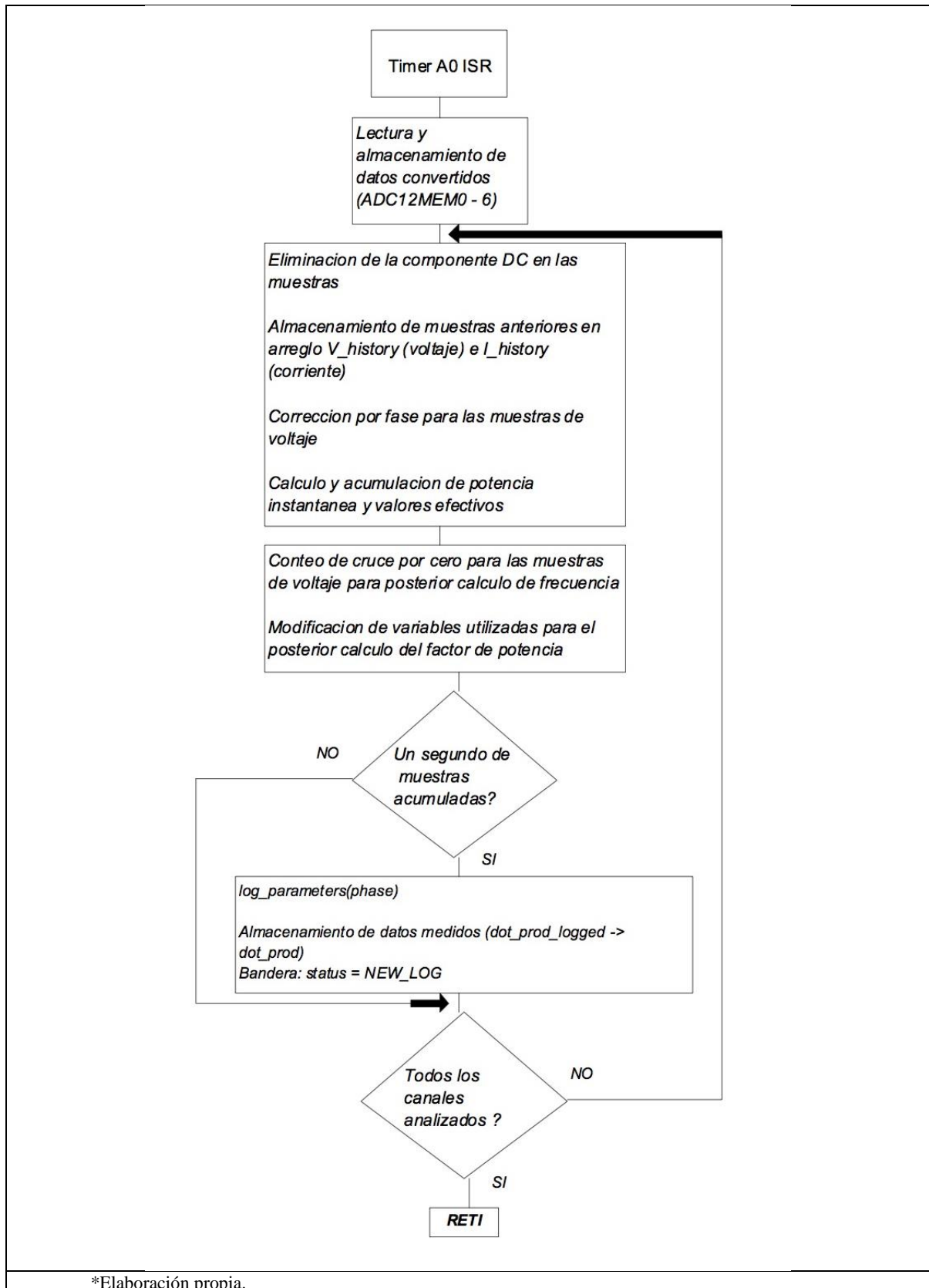


Figura 17: Flujoograma background

Procesamiento y cálculo de variables

Debido a que el proceso de almacenamiento, cálculo y acumulación de datos es lo más importante en el sistema de medición, dentro de esta sección se detalla dicho proceso en un segundo flujograma de las señales pertenecientes a un canal de medición. Este flujograma se presenta a continuación en la *Figura 18*.

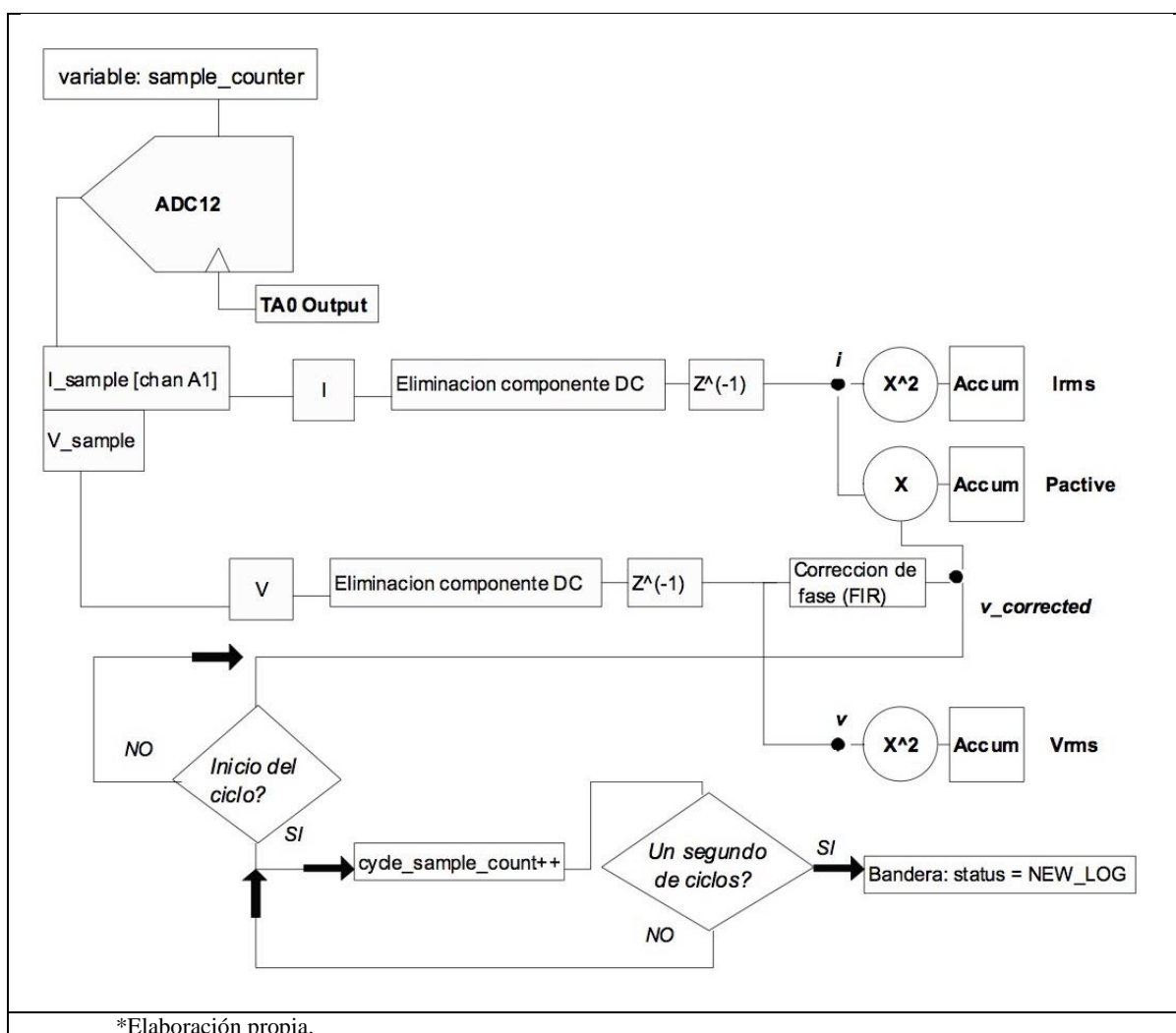


Figura 18: Flujograma de señal por canal

Aquí Z^{-1} representa el almacenamiento de datos anteriores en vectores, X^2 y Accum representan la función matemática utilizada para elevar la muestra al cuadrado y

acumularla en una variable, y X representa la función matemática utilizada para multiplicar la muestra de corriente con la de voltaje.

Conteo de ciclos y medición de frecuencia

Por una parte, en el proceso de conteo se define un contador de ciclos que indica el número de muestras que se han tomado hasta el momento. Cuando se obtiene aproximadamente un segundo de muestras se notifica a foreground.c a través de la bandera status. El número de muestras tomadas en un segundo se encuentra limitado por este contador de ciclos de manera que no se tomen muestras innecesarias por cada segundo. Para la medición de frecuencia por otra parte, se utiliza una interpolación lineal entre las muestras de voltaje que cruzan el valor cero sin tomar en cuenta las muestras cercanas el punto de cruce. Esta interpolación se muestra gráficamente en la *Figura 19*. Debido a que picos de voltaje pueden alterar la distinción entre puntos de cruce, se utiliza en el algoritmo implementado un factor de comprobación de picos el cual descarta la muestra analizada si la diferencia entre ésta y la muestra anterior es mayor al factor.

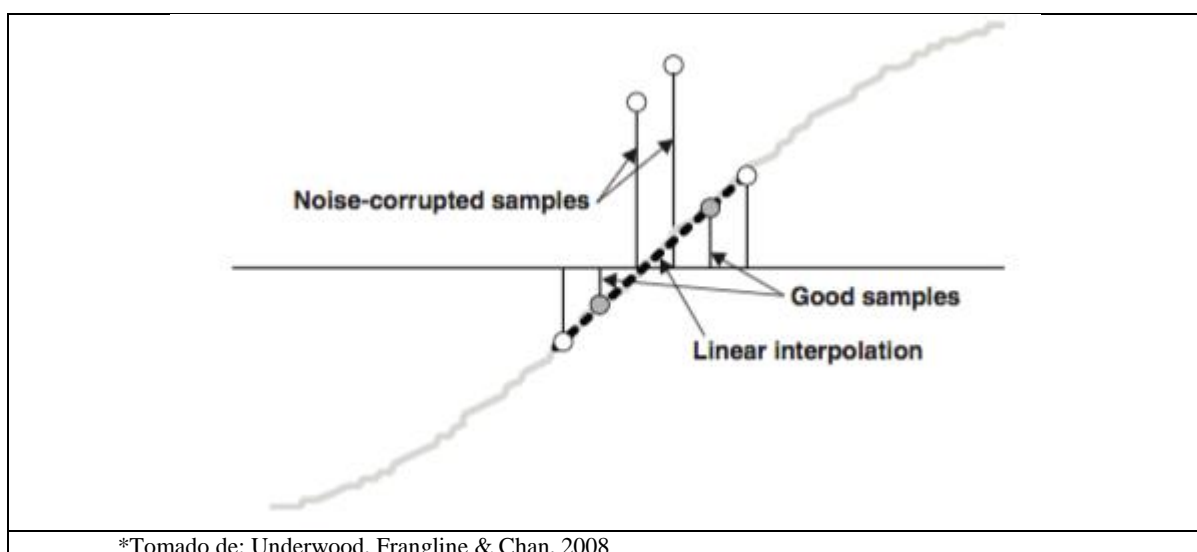


Figura 19: Interpolación lineal

Corrección de fase: Filtro FIR

El principal problema de un sistema de medición basado en el procesamiento digital de datos es que existe un desfase entre la muestra análoga y la muestra digital. Este desfase es causado principalmente por el módulo ADC12, el cual, al tener solo un bloque de muestreo y conversión, inserta un retardo mínimo entre las muestras resultantes. La consecuencia de este retardo generado es un porcentaje de error final al momento de calcular la potencia. Por eso, para completar la estabilidad y confiabilidad en la medición de la potencia instantánea, se realiza una compensación de fase entre estas dos muestras. Para ello, se implementa un filtro FIR de orden cero que provee un retardo en adelante o retraso de media muestra de medición aplicado al voltaje. Esto se presenta en la *Figura 20*.

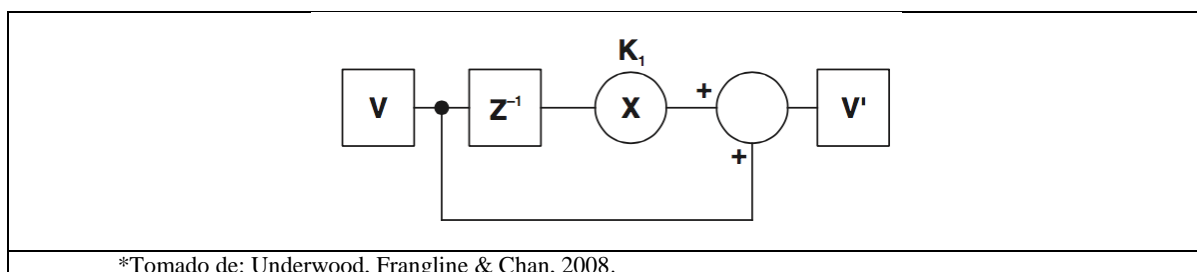


Figura 20: Filtro FIR de orden cero

Como menciona el autor Tam (2001), la implementación de este tipo de filtro se la realiza en base a la ecuación (41).

$$V'[n] = V[n] + \beta V[n - 1] \quad (41)$$

Aquí V es un vector de muestras de voltaje y β es el coeficiente de corrección que depende tanto del ángulo de corrección como de la frecuencia de la onda fundamental. Debido a que el algoritmo requiere de una tabla de filtros FIR para una onda fundamental de 60 [Hz], todas las ecuaciones descritas por Tam se implementaron en un script de Matlab que se presenta en el Anexo E1.1.

Proceso de foreground del sistema (foreground.c)

El diagrama de bloques que se presenta en la *Figura 21* resume el algoritmo de foreground.c del sistema implementado. Posterior a este diagrama se detalla los procesos de mayor relevancia.

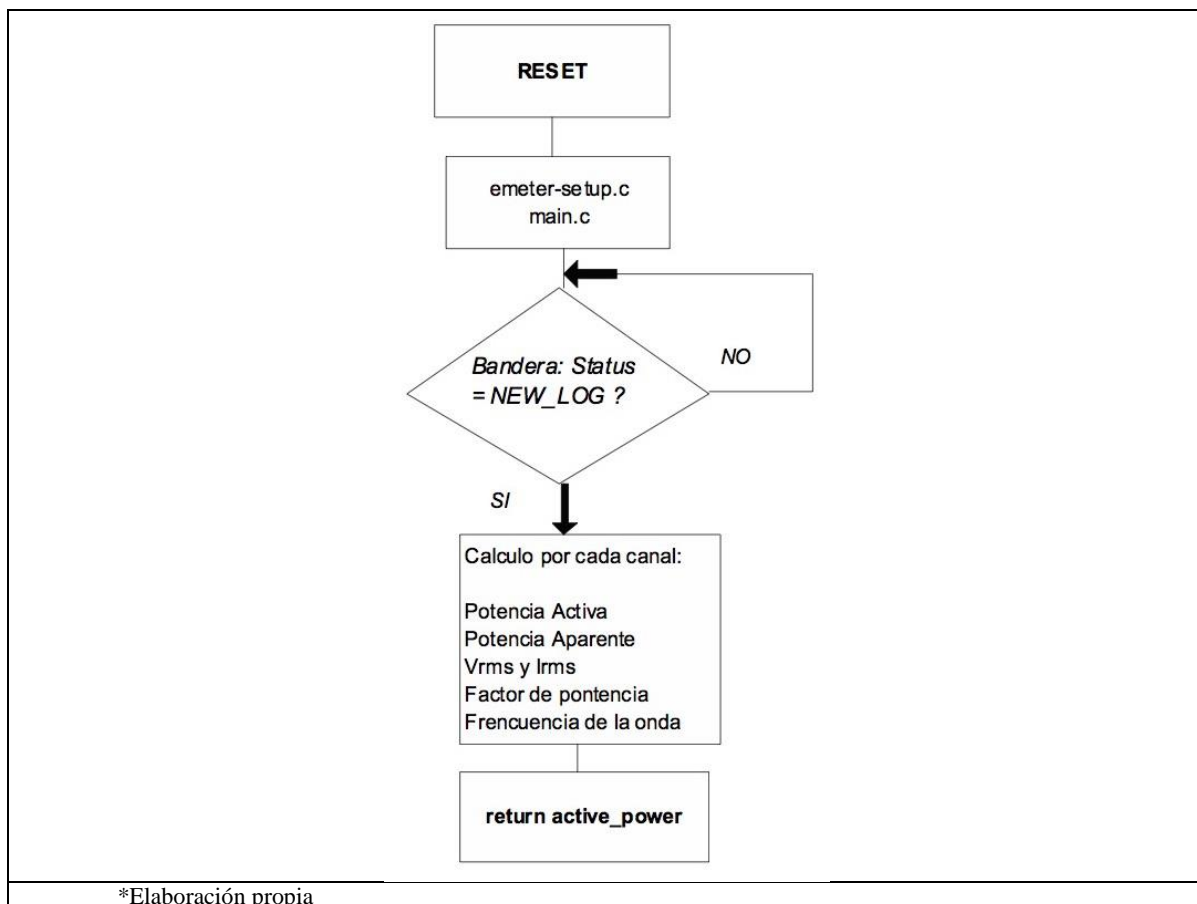
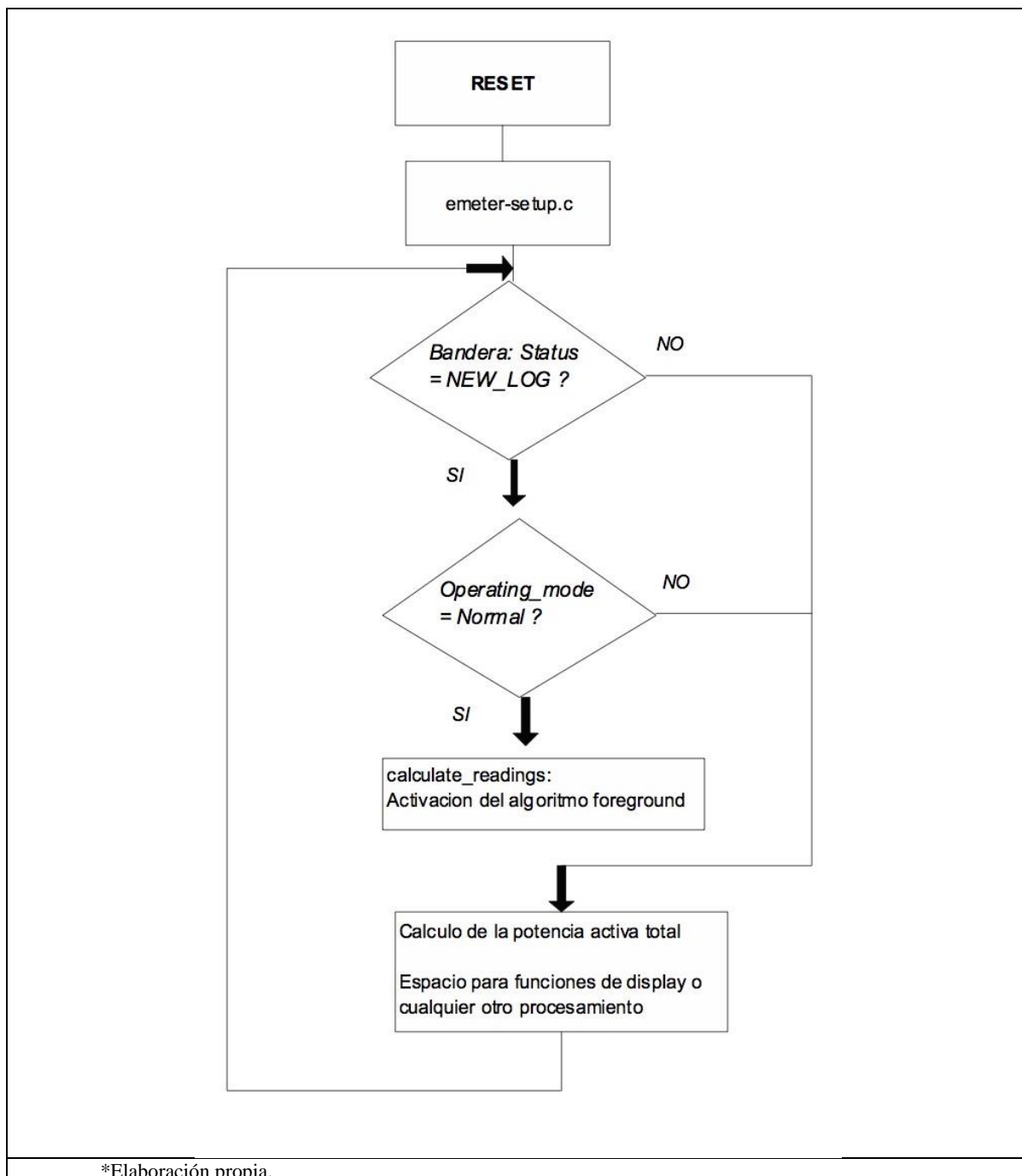


Figura 21: Flujograma Foreground

Como se puede observar, este algoritmo es el encargado del cálculo de todas las variables ya mencionadas. Dicho cálculo se basa en la teoría presentada en el anterior capítulo, desde la ecuación (13) hasta la ecuación (42). Cabe mencionar que los factores de escalamiento utilizados se encuentran declarados en el archivo de cabecera metrology-parms.h. Estos transforman el dato medido por foreground.c a un dato real del sistema de medición codificado como un número de 32 bits, en el cual los primeros 16 codifican la

parte entera y los otros 16 codifican la parte decimal. Sin embargo, para el caso del factor de potencia el dato medido es el dato real por un factor de 10000, y para el caso de la frecuencia el dato medido es el dato real por un factor de 100.

Lazo principal (main) del sistema (main.c)



*Elaboración propia.

Figura 22: Flujograma main

El algoritmo de main.c del sistema implementado se resume en el diagrama de bloques presentado en la *Figura 22*.

Parámetros de calibración

Para concluir con la implementación en software del sistema de medición propuesto, se implementaron algunos parámetros de calibración para reducir el error porcentual en el cálculo de las variables eléctricas. Este error existe debido a ruido en el sistema, impedancia de los cables [jumpers] utilizados, entre otros factores físicos de la tarjeta de adquisición de datos. Estos parámetros, definidos en el archivo metrology-parms.h, son:

- V_AC_OFFSET para todos los canales de medición.
- I_AC_OFFSET para todos los canales de medición.
- DEFAULT_BASE_PHASE_CORRECTION para todos los canales de medición.

Todos estos parámetros mencionados deben ser comprobados y modificados según los resultados de calibración del sistema de medición.

Capítulo 5:

Transmisión Inalámbrica

Fundamentos de la red implementada

Una vez calculadas las variables de medición mencionadas en los anteriores capítulos, lo que se desea a través del trabajo expuesto aquí es la correcta transmisión inalámbrica de las mismas hacia un concentrador central el cual sea capaz de recibir y presentar estos resultados en una interfaz gráfica [GUI]. Por eso, el presente capítulo tiene como objetivos enseñar las características de funcionamiento y las configuraciones realizadas en el módulo de radio frecuencia presente en la tarjeta EM430F6137RF900. Así mismo, se resumen los algoritmos diseñados e implementados en programación de alto nivel para el funcionamiento de la red inalámbrica propuesta. Para la programación se utiliza, al igual que el capítulo anterior, el programa Code Composer Studio conectado a un MSP430-USB-Debug-Interface.

Estructura y protocolos de comunicación implementados

Todo sistema de comunicación nace de un modelo de transmisión. Por esta razón, en esta sección se detallan el modelo de transmisión por capas implementado y los modelos de enrutamiento y de acceso al medio utilizados para asegurar el correcto funcionamiento de la red inalámbrica diseñada.

Modelo de transmisión

Partiendo del modelo OSI detallado en el capítulo dos, se propone un modelo de transmisión basado en una capa de abstracción de hardware. Este modelo se presenta en la

Figura 23. Como se puede observar, existe una primera capa de Hardware en la cual se encuentra toda la estructura física de medición y adquisición de señales. Seguida de esta capa, se tiene a la capa de abstracción de hardware [HAL] en la cual se realizan todos los procesos de digitalización de datos, procesamiento y transmisión inalámbrica de los mismos utilizando los registros y funcionalidades del sistema CC430. Finalmente, la última capa, la capa de aplicación, involucra una interfaz gráfica [GUI] implementada en un concentrador de datos como se propone en el siguiente capítulo.

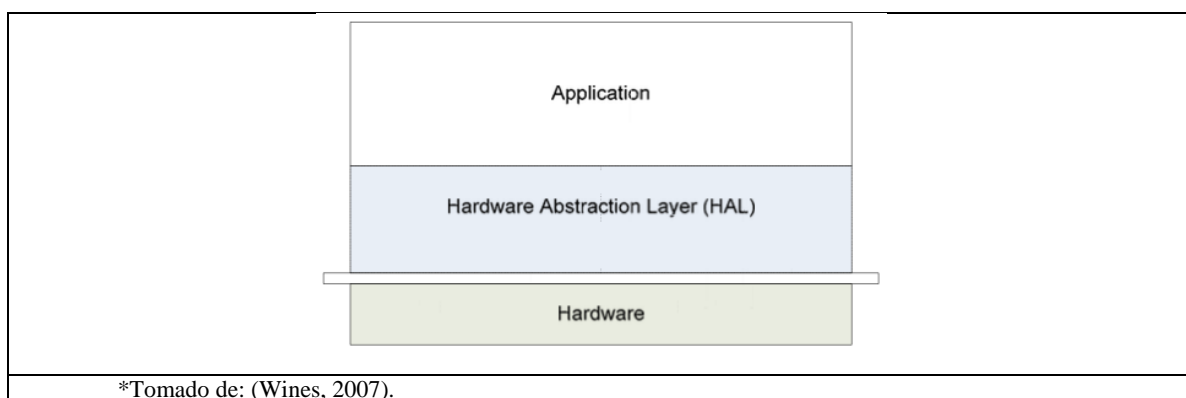


Figura 23: Modelo HAL

Protocolo de acceso al medio (MAC)

El protocolo de acceso al medio (MAC) propuesto se basa en el estándar IEEE 802.5 mencionado en los fundamentos. Este sistema de tipo “listen-before-talk” maneja un vector de direcciones el cual se convierte en el “token” de transmisión. Es decir, al inicio de la comunicación el dispositivo con menor dirección física tiene la primera prioridad de transmisión. Al finalizar la misma, este “token” es pasado al dispositivo con la siguiente dirección numérica para que pueda utilizar el medio mientras que los otros dispositivos solo se dedican a escuchar y esperar. Una vez que todos los dispositivos dentro de la red hayan transmitido su información, se reinicia el valor de este “token”, lo que permite al primer dispositivo utilizar nuevamente el medio de comunicación.

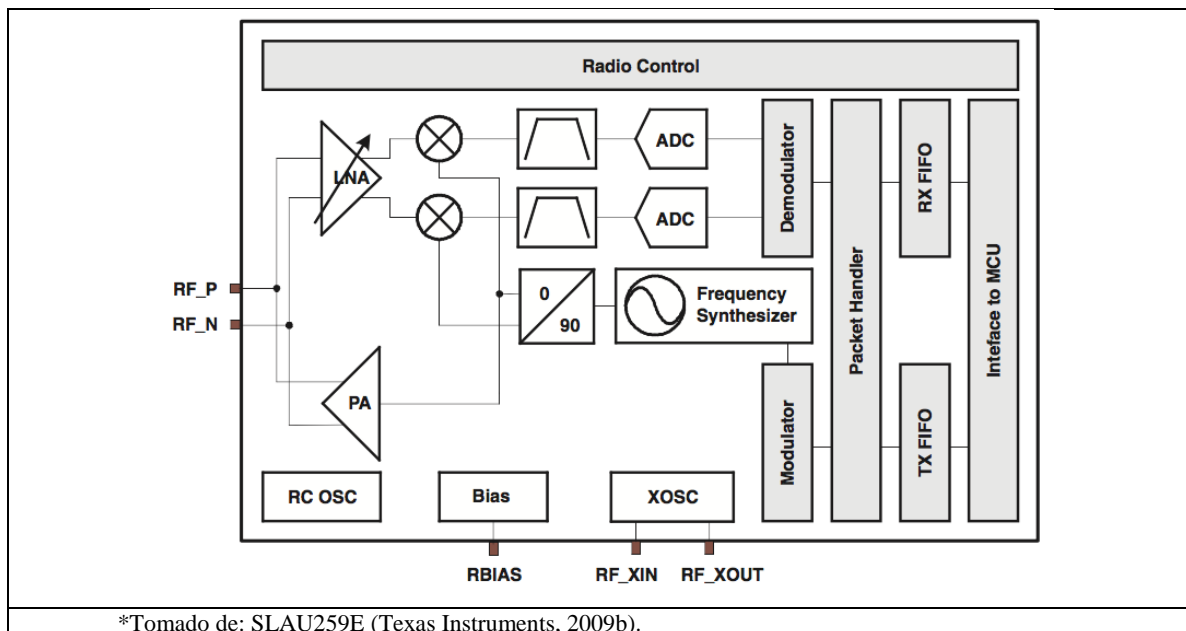
Protocolo de enrutamiento

El protocolo de enrutamiento utilizado se basa en el protocolo de internet UDP ya que en éste no se realiza ninguna comprobación del paquete enviado. El protocolo implementado utiliza direcciones lógicas estáticas en cada uno de los dispositivos dentro de la red así como en el concentrador o Gateway de destino. Es importante recalcar que a partir del modelo de transmisión propuesto y los protocolos de comunicación implementados la red inalámbrica diseñada se convierte en una red que es:

- Estable: al mantener sus características de funcionamiento a pesar de una posible variación en la carga de medición.
- Escalable: al permitir la participación de más dispositivos dentro de la red siempre y cuando compartan el mismo modelo y protocolo de transmisión.
- Resiliente: al mantener el funcionamiento de la red a pesar de que uno o más integrantes dejen de transmitir.

Módulo de radio frecuencia CC1101 (RF1A)

El módulo de radio frecuencia RF1A es el encargado de integrar al núcleo inalámbrico CC1101 al sistema de la tarjeta EM430F6137RF900. Este núcleo que opera a través del transceptor CC1101, se lo utiliza frecuentemente en aplicaciones de bajo consumo de potencia. Estas aplicaciones son implementadas en bandas de transmisión industriales, científicas o médicas (ISM) o en bandas de corto alcance (SRD) las cuales pueden ser fácilmente programables como se detallará en la siguiente sección. Para resumir de forma general el funcionamiento de este módulo se presenta *Figura 24* su diagrama de bloques simplificado.



*Tomado de: SLAU259E (Texas Instruments, 2009b).

Figura 24: Diagrama de Bloques Módulo RF1A

Como se puede observar en el diagrama presentado, el módulo RF1A contiene un receptor de frecuencia intermedia (IF). Para el proceso de recepción, la señal de radio frecuencia recibida (RF) es amplificada por un amplificador de bajo ruido (LNA) y posteriormente convertida en señales de cuadratura (I y Q) hacia el receptor IF. En este receptor, las señales (I/Q) son digitalizadas por un convertor análogo digital (ADC). Una vez obtenidas estas señales, se realizan en un formato digital algunos procesos tales como: control de ganancia, filtrado por canal de transmisión, demodulación y sincronización de paquetes (Texas Instruments, 2009b).

Para el proceso de transmisión se incluye un sintetizador de frecuencia. Este sintetizador contiene un oscilador controlado por voltaje (VCO) y un desfasador de 90° para generar las señales de cuadratura (I/Q). Un oscilador de cristal de 26 [MHz] es utilizado como referencia para el sintetizador de frecuencia así como para los relojes de

operación de los módulos de conversión análoga digital (Texas Instruments, 2009b). Para concluir, las principales características de este módulo se resumen en la *Tabla 7*.

Tabla 7: Características Módulo RF1A

Característica	Descripción
Frecuencias de transmisión	<ul style="list-style-type: none"> • 300 – 348 [MHz] • 389 – 464 [MHz] • 779 – 928 [MHz]
Velocidad de transmisión de datos	<ul style="list-style-type: none"> • 0.8 – 500 [kBaud]
Potencia de salida	<ul style="list-style-type: none"> • Se puede programar una potencia de salida de hasta 10 [dBm] para todas las frecuencias de transmisión
Modulación de datos	<ul style="list-style-type: none"> • FSK • GFSK • MSK • OOK • ASK
Sistemas de transmisión y recepción	<ul style="list-style-type: none"> • Sistema orientado al manejo de paquetes • Sistema de comunicación sincrónica y asincrónica
Indicadores o sistemas adicionales	<ul style="list-style-type: none"> • Sistema de detección del uso del canal de transmisión para protocolos “listen-before-talk” • Implementación de un indicador del estado del enlace (LQI) • Implementación de un indicador digital de la fuerza en la señal recibida (RSSI)

Manejo y funcionamiento del módulo RF1A

El módulo de radio frecuencia RF1A está compuesto principalmente por su núcleo y su interfaz. En su núcleo se encuentran todos los registros de funcionamiento y desde su interfaz se maneja al núcleo a través de una programación adaptada. La comunicación

entre la interfaz y el núcleo toma lugar en base a la utilización de canales lógicos tal como se presenta en la *Figura 25*. En esta imagen se puede observar que toda instrucción dada al núcleo resulta en una actualización del estado hacia la interfaz. De igual forma, se puede observar que toda información enviada al núcleo resulta en un dato en byte (8-bits) o en palabra (16-bits) devuelto hacia la interface.

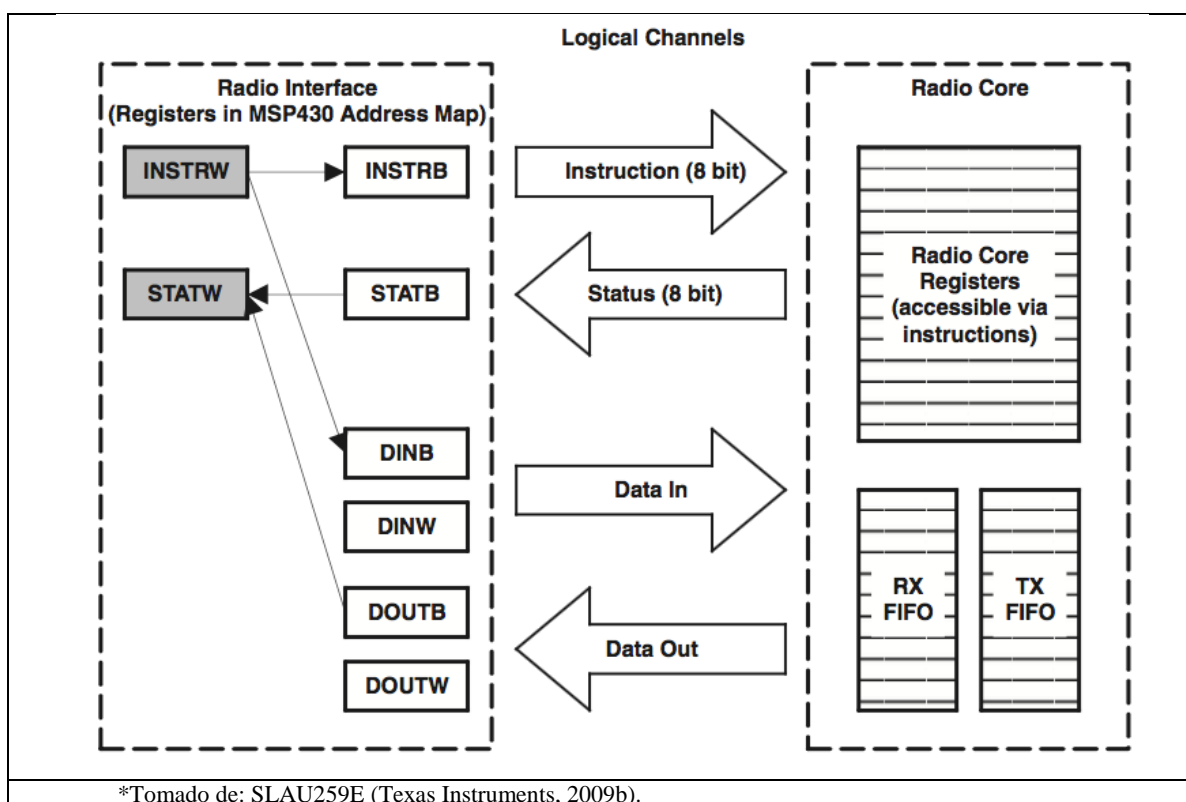


Figura 25: Comunicación en el módulo RF1A

De la figura anterior se puede deducir que comunicación entre la interfaz y el núcleo del módulo RF1A depende de los diferentes registros de instrucción y de datos presentes en el módulo. Por un lado, los registros de instrucción RF1AINSTR_xW/xB permiten controlar el núcleo del módulo mientras que por otro lado, los registros de datos RF1ADINW/B permiten proveer mayor información en la instrucción o comando deseado. A su vez, los registros de datos también pueden ser utilizados para pasar los parámetros o

información necesaria al núcleo del RF1A para su posterior procesamiento. Es importante considerar que la forma correcta de acceso a y escritura de estos diferentes registros puede ser bajo tres metodologías implementadas sea de forma individual o grupal:

1. Acceso directo a los registros de control.
2. Acceso directo a las banderas de interrupción del módulo.
3. Utilización de las rutinas de interrupción (ISR).

Fundamentos operativos del núcleo RF1A en la red inalámbrica implementada

En base a las características expuestas en la *Tabla 7*, en esta sección se explica sobre la configuración específica de los módulos operativos de mayor relevancia en la red inalámbrica implementada.

Empaquetamiento FIFO de la información

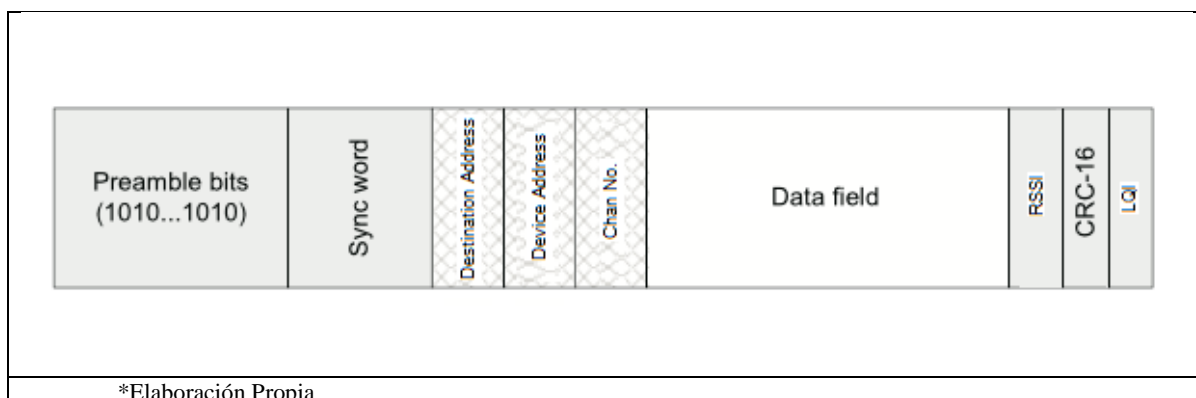


Figura 26: Encapsulamiento Implementado

Como ya se ha mencionado en los anteriores capítulos, el formato general de encapsulamiento de la información se representa como se muestra en la *Figura 2*. Sin embargo, en la red inalámbrica implementada se agregan ciertos parámetros a este formato, los cuales se presentan en la *Figura 26*. Cada una de estas secciones de empaquetamiento cumple una función en específico dentro del proceso tanto de transmisión así como de recepción de datos. El patrón de preámbulo es una secuencia

alterna de unos y ceros que es enviada el momento en el que se habilita la funcionalidad de transmisor en el módulo RF1A. Este preámbulo es enviado constantemente hasta el momento en que se detecta información en el espacio de almacenamiento TX FIFO. En ese instante, se detiene el envío del preámbulo y se envía la palabra de sincronización contenida en los registros SYNC1/0.

Posterior al envío de los registros de sincronización, se inserta en el campo de datos bytes de información e identificación tales como:

- Dirección de destino: byte que codifica la dirección del dispositivo al cual se quiere transferir la información.
- Dirección del dispositivo transmisor: byte que codifica la dirección del dispositivo que transmite la información como método de identificación dentro de la red.
- Número de canal de medición: byte que codifica el canal análogo digital del cual provienen todas las variables de medición transmitidas.

La información transmitida y codificada dentro del campo de datos [Data Field] se divide y estructura de la siguiente forma:

- Bytes [8:3]: estampilla de tiempo que codifica el día, mes, año, hora, minuto y segundo en el cual se realizó dichas mediciones.
- Bytes [12:9]: potencia activa medida codificada en 16-bits parte entera y 16-bits parte decimal.
- Bytes [16:13]: potencia aparente medida codificada en 16-bits parte entera y 16-bits parte decimal.
- Bytes [20:17]: voltaje efectivo medido codificado en 16-bits parte entera y 16-bits parte decimal.

- Bytes [24:21]: corriente efectiva medida codificada en 16-bits parte entera y 16-bits parte decimal.
- Bytes [26:25]: factor de potencia medido.
- Bytes [28:27]: frecuencia de la onda de voltaje medida.

Finalmente, en la cola del paquete de información se inserta el identificador digital de la fuerza de la señal recibida (RSSI), la cadena de control de datos para una correcta recepción (CRC) y el identificador de estado de enlace (LQI).

Indicador de la fuerza en la señal recibida (RSSI)

El indicador digital de la fuerza en la señal recibida (RSSI) es una estimación del nivel de potencia en la señal, medido en [dBm] y dentro del canal de transmisión escogido (Texas Instruments, 2009b). En modo de recepción [RX], este valor puede ser leído constantemente para así poder reconocer cambios en el nivel de potencia dentro del canal de transmisión.

Detección del uso del canal (CCA)

El identificador CCA es frecuentemente utilizado por el núcleo RF1A para reconocer si el medio de transmisión se encuentra libre u ocupado. Este identificador se puede observar a través de la bandera de interrupción RFIFG12, la cual puede ser configurada para activarse en el momento en el que se está recibiendo un paquete o en el que el nivel RSSI del medio se encuentra bajo el límite preestablecido.

Indicador de estado de enlace (LQI)

El identificador LQI es una métrica que describe la calidad de la señal transmitida y la estimación de la facilidad en la que la señal de recepción puede ser demodulada y acumulada.

Programación de la red inalámbrica en banda ISM

El desarrollo de los algoritmos que se presentan a continuación tiene como sustento teórico un trabajo previo realizado por Texas Instruments. “CC430 RF Examples” es una nota técnica en la que se disponen todos los códigos desarrollados para cada modo de comunicación disponible en los sistemas CC430 (Morales & Dang, 2012). Algunos de los módulos expuestos aquí son: comunicación por paquetes con tamaño de almacenamiento menor o mayor que los paquetes tipo FIFO [64 bytes], comunicación sincrónica y comunicación asincrónica. Debido a que el microprocesador utilizado para el proyecto de la presente tesis [CC430F6137] forma parte de los sistemas CC430, todo el código desarrollado en esta nota técnica pudo ser utilizado sin ningún proceso de adaptación necesario. Adicionalmente al código tomado de la nota técnica, el desarrollo de la programación de la red inalámbrica cuenta también con un aporte propio de funciones que se adaptan a la estructura de comunicación necesaria y con subrutinas que implementan los protocolos de enrutamiento y acceso al medio.

Rutinas y algoritmos

En la presente sección se exponen las rutinas y algoritmos para utilizar el núcleo del módulo RF1A y transmitir los resultados de medición obtenidos en las distintas variables de voltaje, corriente y potencia previamente calculadas. Estos algoritmos, además de codificar los procesos de transmisión, establecen el funcionamiento de los protocolos de comunicación implementados. La programación de todas las rutinas y subrutinas utilizadas se las realiza en lenguaje C. Es importante mencionar que todos los algoritmos presentes en esta sección deben formar parte y funcionar dentro del código desarrollado en el capítulo anterior. Por esta razón, la metodología de implementación del módulo de

comunicación inalámbrica dentro del sistema de medición propuesto se realizó en base a dos procesos:

1. Inserción de nuevos archivos de cabecera con la declaración de las distintas funciones necesarias para el correcto funcionamiento del módulo RF1A.
2. Inserción de nuevas subrutinas dentro de los archivos main.c, emeter-setup.c y foreground.c para la transmisión de las variables de medición y la implementación del protocolo de comunicación.

Estas metodologías de implementación serán detalladas a continuación.

Ficheros o archivos de cabecera (header file).h

Un resumen de los archivos de cabecera insertados se presenta en la *Tabla 8*.

Tabla 8: Archivos de cabecera módulo RF1A

Archivo de cabecera	Descripción
RF_Toggle_LED_Demo.h	<ul style="list-style-type: none"> • Declaración de las funciones para inicialización del módulo RF1A • Declaración de las funciones para modo transmisión [TX] o recepción [RX] • Declaración de la función para la implementación del sistema “listen-before-talk”
hal_pmm.h	<ul style="list-style-type: none"> • Declaración de las funciones que controlan el nivel de energía necesario para el correcto funcionamiento del microprocesador
RF1A.h	<ul style="list-style-type: none"> • Declaración de la estructura de variables para la configuración del módulo RF1A • Declaración de las funciones para la escritura y lectura del espacio de almacenamiento FIFO TX o FIFO RX

Módulo RF1A dentro de la inicialización del sistema (emeter-setup.c)

En esta sección se explica la subrutina insertada dentro del archivo emeter-setup.c. Esta subrutina configura e inicializa los registros asociados al módulo RF1A. A pesar de que el algoritmo de inicialización del sistema ya se resume en el diagrama de bloques de la *Figura 15*, en el diagrama de bloques que se presenta en la *Figura 27* se detalla específicamente la subrutina insertada dentro del código de inicialización. Los procesos o subrutinas de mayor relevancia encontrados en este diagrama se detallan a continuación.

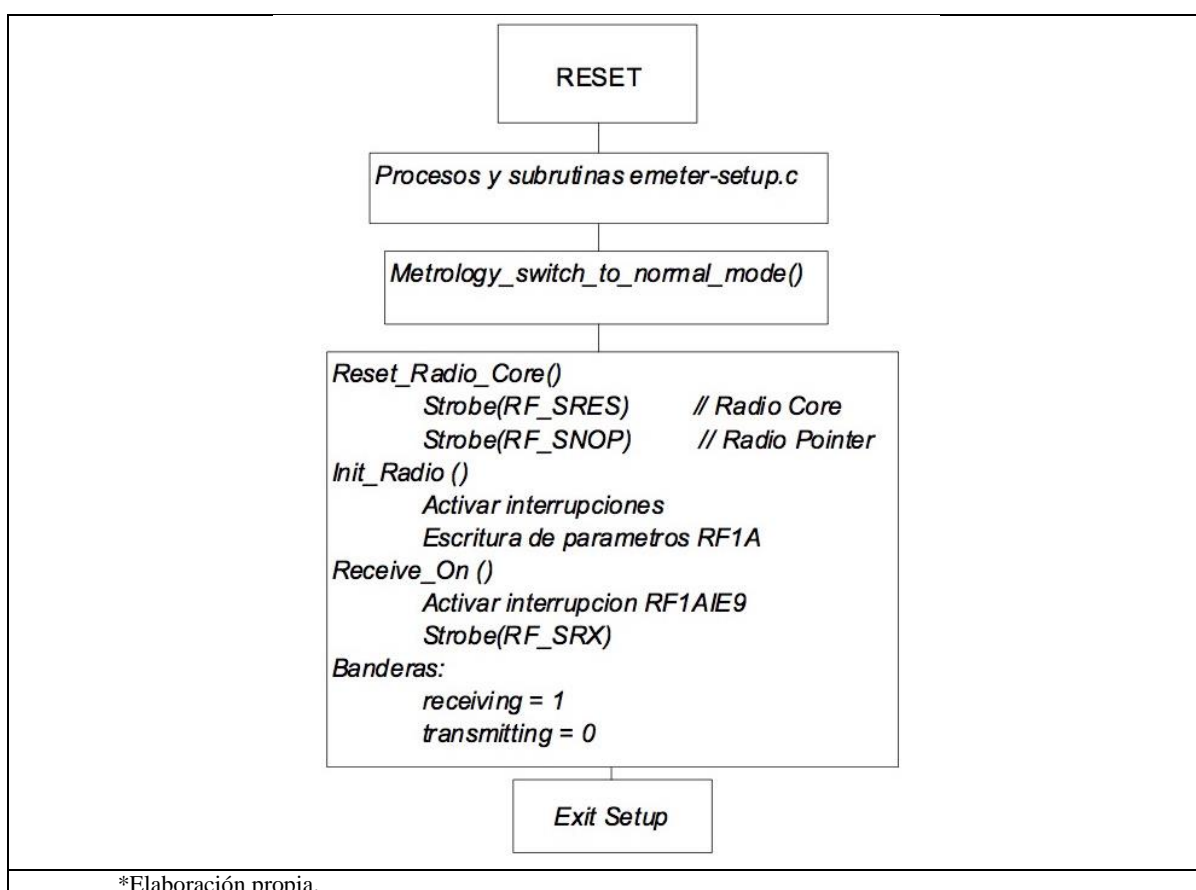


Figura 27: Flujo de Inicialización (setup) con módulo RF1A

Comandos y procesos específicos del módulo RF1A

En la *Figura 27* se puede observar que se utiliza la función “Strobe()” la cual permite configurar el núcleo del módulo RF1A acorde a la instrucción utilizada. En este caso se utilizan las instrucciones:

- SRES: Reinicio del núcleo
- SNOP: Sin operación. Se utilizada para leer el byte de estado
- SRX: Habilitación del modo receptor [RX]

Todas las instrucciones disponibles se encuentran en la Tabla 25-6 del documento SLAU259E (Texas Instruments, 2009b).

Escritura de parámetros para la configuración del módulo RF1A (RFRegSettings.c)

En la *Tabla 9* que se presenta en seguida, se resumen las configuraciones más importantes realizadas para el módulo RF1A con sus respectivas ecuaciones. La mayoría de los valores utilizados en los registros de configuración son resultado, por un lado, de los parámetros de diseño utilizados en la ficha técnica consultada y, por otro lado, de los valores resultantes al utilizar el programa SmartRF Studio con los parámetros de diseño propuestos.

Tabla 9: Configuración módulo RF1A

Registro	Valores Asociados [hex]	Ecuaciones Asociadas	Resultado
FSCTRL1	$FREQ_{IF} = 0x08$	$f_{IF} = \left(\frac{f_{XT2}}{2^{10}}\right) \cdot FREQ_{IF}$	$f_{IF} = 203[KHz]$ @26[MHz]
FREQ[2:0]	$FREQ[23:0] = 0x216276$	$f_{carrier} = \left(\frac{f_{XT2}}{2^{16}}\right) \cdot FREQ[23:0]$	$f_{carrier} = 867.99[MHz]$ @26[MHz]
MDMCFG4	$CHANBW_E[1:0] = 0x03$ $CHANBW_M[1:0] = 0x00$ $DRATE_E[3:0] = 0x0A$	$BW_{chan} = \frac{f_{XT2}}{8 \cdot (4 + CHANBW_M) \cdot 2^{CHANBW_E}}$	$BW_{chan} = 101.6[KHz]$ @26[MHz]
MDMCFG3	$DRATE_M[7:0] = 0x83$	$R_{DATA} = \frac{(256 + DRATE_M) \cdot 2^{DRATE_E}}{2^{28}} \cdot f_{XT2}$	$R_{DATA} = 38.38[KBaud]$ @26[MHz]
MDMCFG2	$MDMCFG2[7:0] = 0x97$	N/A	• Modulación GFSK • Sincronización: 30/32+CCS
MDMCFG1	$MDMCFG1[7:0] = 0x22$	N/A	• 4-bytes de preámbulo
DEVIATN	$DEVIATION_E[2:0] = 0x03$ $DEVIATION_M[2:0] = 0x04$	$f_{dev} = \frac{f_{XT2}}{2^{17}} \cdot (8 + DEVIATION_M) \cdot 2^{DEVIATION_E}$	$f_{dev} = 19.04[KHz]$ @26[MHz]

MCSM1	$MCSM1[7:0] = 0x30$	N/A	<ul style="list-style-type: none"> • Modo IDLE al finalizar modo RX o TX • Activación modo CCA cuando se recibe un paquete o el RSSI está bajo el límite establecido
FIFOTHR	$FIFO_THR[3:0] = 0x07$	N/A	<ul style="list-style-type: none"> • 33-bytes en TX FIFO • 32-bytes en RX FIFO
PKTCTRL1	$PKTCTRL1[7:0] = 0x05$	N/A	<ul style="list-style-type: none"> • Adjuntar indicadores RSSI y LQI • Verificación de direcciones sin dirección de broadcast
PKTCTRL0	$PKTCTRL0[7:0] = 0x04$	N/A	<ul style="list-style-type: none"> • Soporte para cálculo de errores CRC • Tamaño fijo de paquete definido en registro PKTLEN
ADDR	$ADDR[7:0] = 0x0n$	N/A	<ul style="list-style-type: none"> • Dirección del dispositivo dentro de la red (n=dirección)
PKTLEN	$PKTLEN[7:0] = 0x1D$	N/A	<ul style="list-style-type: none"> • Tamaño del paquete de información

Módulo RF1A dentro del proceso de foreground del sistema (foreground.c)

El diagrama de bloques que se presenta en la *Figura 28* resume el algoritmo del sistema implementado insertado dentro de foreground.c. Como se puede observar, existen nuevos elementos dentro del flujograma que cumplen los siguientes objetivos:

- Función `time_stamp`: insertar la estampilla de tiempo en las mediciones realizadas.
- Función `capsule_8bit`: dividir las variables de medición en pedazos de 8-bits ya que es este el formato aceptado por el espacio de almacenamiento TX o RX FIFO.
- Bandera `measurements_ready`: indicar al lazo principal [main.c] que las variables de medición están listas para ser transmitidas.

Posterior a este diagrama se detalla los procesos de mayor relevancia.

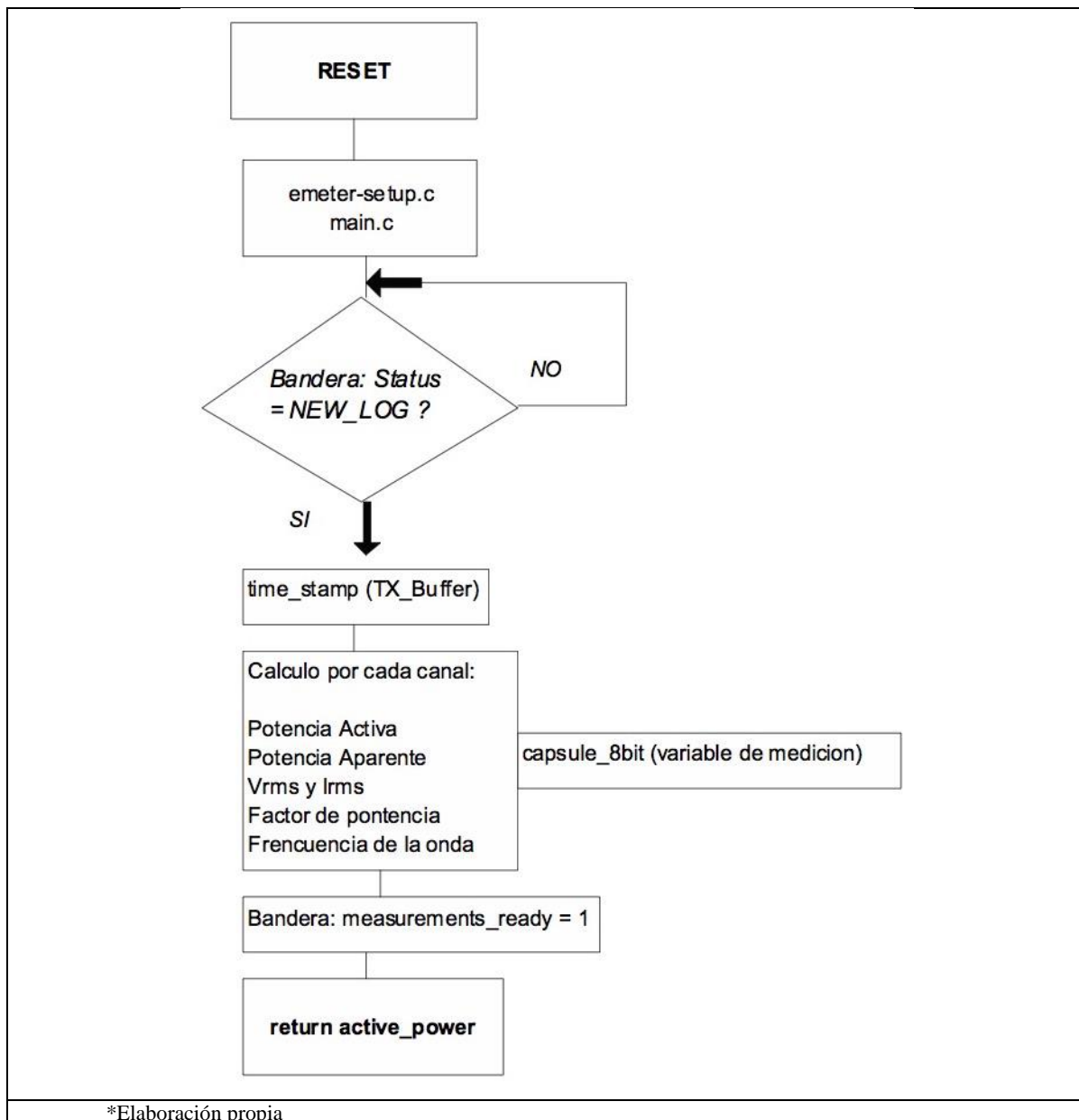


Figura 28: Flujograma Foreground-RF1A

Módulo RF1A dentro del lazo principal (main) del sistema (main.c)

Como se mostró ya, el algoritmo de la rutina main.c se resume en el diagrama de bloques presentado en la *Figura 22*. Sin embargo, al añadir el módulo inalámbrico a este lazo se obtiene el diagrama de bloques presentado en la *Figura 29*.

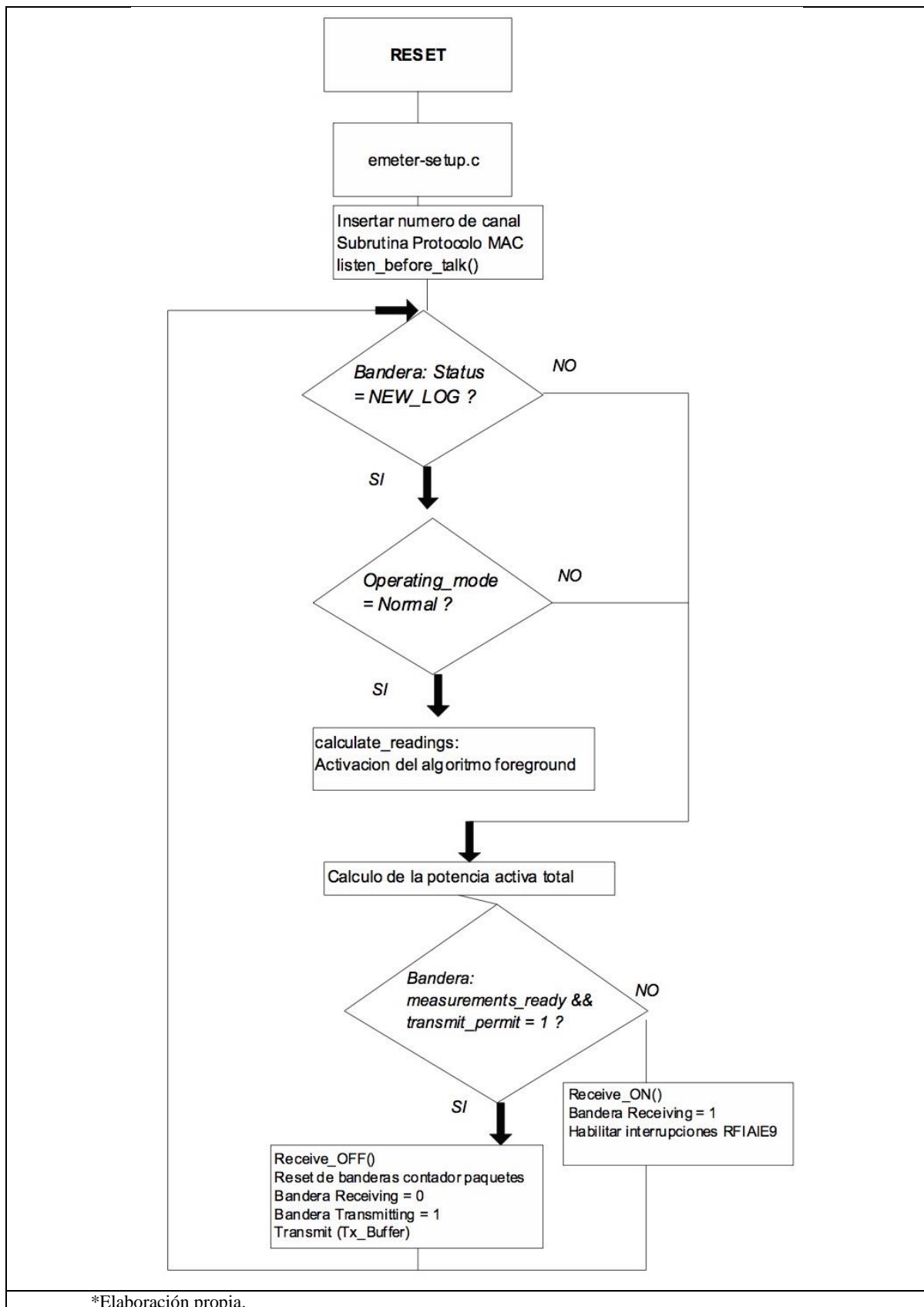


Figura 29: Flujograma main-RF1A

Como se puede observar en el diagrama de bloques, existen nuevos elementos dentro del flujograma que cumplen los siguientes objetivos:

- Subrutina Protocolo MAC y función listen-before-talk(): comprueban y actualizan la pertenencia del “token” dentro de la red para poder así permitir [bandera: transmit_permit = 1] o negar [bandera: transmit_permit = 0] la transmisión de datos del dispositivo.
- Funciones Receive_ON/OFF (): activar o desactivar el modo de recepción RX del sistema.
- Sistema de contador de paquetes: conjunto de variables que controlan el número de paquetes recibidos o transmitidos en el sistema. Estas variables se las modifica dentro de la rutina de interrupción tanto de la antena CC1101 como del reloj [emeter-rtc.c].
- Función Transmit(TX_Buffer): transmitir la cadena de datos insertada en el espacio de almacenamiento TX FIFO.

Capítulo 6:

Concentración y Presentación de Datos

Fundamentos en la recepción y presentación de datos

Finalizado el proceso de medición y transmisión de datos, se completa el funcionamiento del DAQ con el desarrollo de un sistema de recepción y presentación de datos. Este sistema, se compone de una parte física y de una interfaz gráfica [GUI]. La parte física está basada en el funcionamiento de la tarjeta CC1111 USB Dongle que actúa como antena receptora de datos. La interfaz gráfica es desarrollada en LabVIEW con el propósito de conectarse, mediante comunicación USB-Serial con la tarjeta receptora y presentar los datos de medición en una plataforma visual. En el presente capítulo, se detalla las características implementadas en cada una de estas dos partes constituyentes del sistema de recepción mencionado.

CC1111 USB Dongle

CC1111 USB Dongle es un dispositivo que forma parte de los sistemas de radio frecuencia de baja potencia de Texas Instruments. El núcleo funcional de esta tarjeta es el chip CC1111F32 que, al igual que el módulo CC1101 utilizado en las tarjetas de transmisión, funciona como un transceptor funcional en bandas ISM o SRD. Este transceptor es compatible con los transceptores utilizados en la transmisión. En consecuencia, la tarjeta CC1111 USB Dongle puede ser utilizada en el presente trabajo como receptor de los datos transmitidos en el aire. Es importante mencionar que la programación de este transceptor varía un poco con la programación de los transmisores debido a la diferencia entre las arquitecturas de los microcontroladores utilizados.

Hardware de Recepción

El microprocesador CC1111F32 contiene varias características de funcionamiento divididas y comandadas a través de la configuración de algunos módulos. Sin embargo, debido a que solo se necesita la recepción inalámbrica de datos y la transmisión de los mismos mediante una comunicación USB-Serial, en esta sección se detallan únicamente las características del controlador USB y las del módem en banda base inferior a 1 [Ghz].

Controlador USB

El módulo USB del microprocesador CC1111F32 permite la comunicación serial con cualquier computadora o equipo calificado para soportar dicha funcionalidad USB. Este controlador es capaz de monitorear la actividad en el puerto USB del dispositivo y administrar el intercambio de paquetes de información. El microprocesador CC1111F32 actuará siempre como esclavo en la comunicación y solo envía o recibe un paquete si el anfitrión [PC] manda una solicitud [token]. Como estándar dentro de la comunicación USB, se denomina proceso IN a la transferencia de datos hacia el computador [PC] y se denomina proceso OUT como la transferencia de datos hacia el microprocesador CC1111F32.

La comunicación de este módulo se resume en el diagrama de la *Figura 30*. Como se puede observar, el controlador se compone de una interfaz física [USB PHY], una interfaz serial [USB SIE] y un registro de memoria. La interfaz física contiene los controladores de entrada y salida de datos, la interfaz serial es la encargada de controlar los paquetes de información desde y hacia cada controlador de punto final [EPx] y el registro es el responsable de conectar al controlador con todo el sistema del microprocesador. La

comunicación USB puede ser resumida en tres tipos de transacciones que definen tanto la transmisión como la recepción de datos.

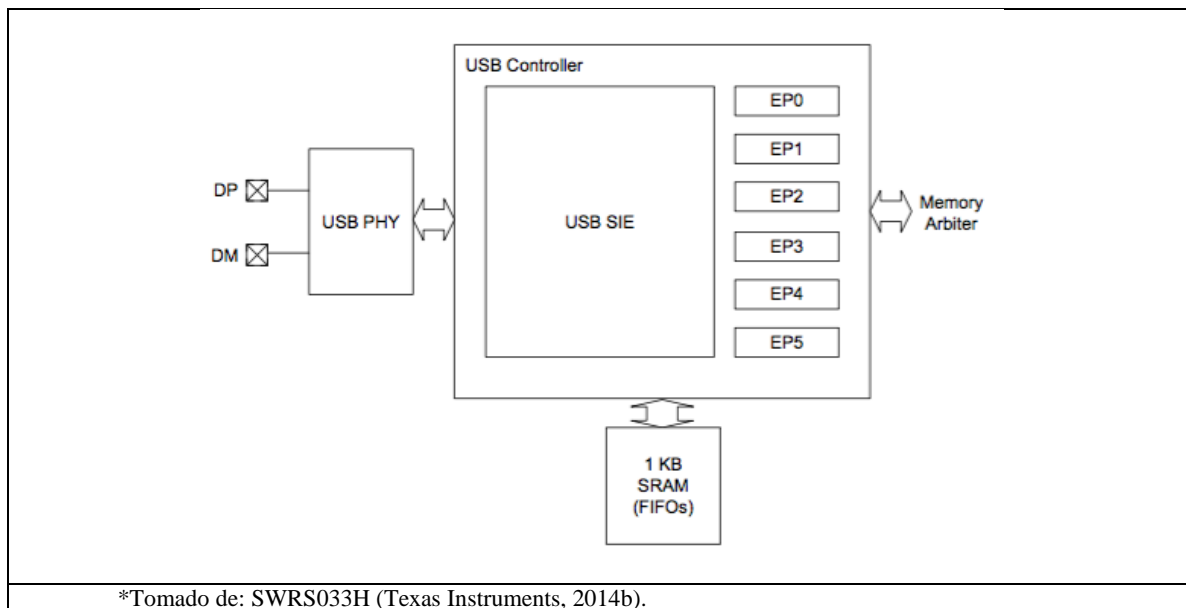


Figura 30: Controlador USB

Transacción SETUP

La transacción de configuración [SETUP] es el primer tipo de intercambio de información en toda comunicación USB. Para que una transacción SETUP sea exitosa, debe componerse de tres paquetes secuenciales: paquete token, paquete de datos y un paquete de negociación [handshake]. Si estos paquetes son enviados y recibidos correctamente, se genera una interrupción en el controlador de punto final cero [EP0]. Este punto es donde se examina el contenido del paquete y se habilita el módulo USB para cualquiera de las siguientes transacciones.

Transacción IN (Estado TX)

Las transacciones IN son utilizadas para enviar paquetes de datos hacia el anfitrión [PC]. Para que una transacción IN sea exitosa, debe componerse de los mismos tres paquetes que se compone una transacción SETUP. Para el caso de una transmisión mayor

a 32 bytes, los paquetes son separados en múltiplos de 32. En este proceso, el primer paquete es transferido al espacio de almacenamiento FIFO EP0, seguido por los paquetes restantes. Esta transmisión se realiza hasta el aviso del envío del último paquete de datos [inferior a 32 bytes].

Transacción OUT (Estado RX)

Las transacciones OUT son utilizadas para recibir información enviada desde el anfitrión [PC]. Esta transacción comparte dos características con la transacción IN: la necesidad de tres paquetes secuenciales para una transacción exitosa y la separación de paquetes grandes en múltiplos de 32 bytes. En este proceso, una interrupción en el controlador de punto final cero [EP0] es generada cada vez que se recibe un paquete de datos. El momento en el que se recibe el último paquete de datos [inferior a 32 bytes], se notifica la culminación del proceso de comunicación a través de la configuración del registro USBCS0.DATA_END.

Es importante recalcar que no todos los controladores de punto final tienen el mismo tamaño de almacenamiento FIFO. Como se puede observar en la figura *Figura 30*, existen cinco diferentes controladores de punto final. Por lo tanto, existen cinco diferentes tamaños de almacenamiento FIFO definidos por la ecuación (42).

$$FIFO = 32 \cdot 2^{N_{ep}-1} [bytes] \quad (42)$$

Aquí N_{ep} es el número de controlador de punto final.

Módulo RF sub 1[GHz]

El módulo de radio frecuencia es el encargado de integrar al núcleo inalámbrico CC1111F32 al sistema de la tarjeta CC1111 USB Dongle. Para resumir de forma general

el funcionamiento de este módulo se presenta en la *Figura 31* su diagrama de bloques simplificado. Como se puede observar, el módulo RF contiene un receptor de frecuencia intermedia [IF]. Para el proceso de recepción, la señal de radio frecuencia recibida [RF] es amplificada por un amplificador de bajo ruido [LNA] y posteriormente convertida en señales de cuadratura [I y Q] utilizadas por el receptor IF. En este receptor, las señales [I/Q] son digitalizadas por un conversor análogo digital [ADC]. Una vez obtenidas estas señales, se realizan en un formato digital algunos procesos tales como: control de ganancia, filtrado por canal de transmisión, demodulación y sincronización de paquetes (Texas Instruments, 2014b).

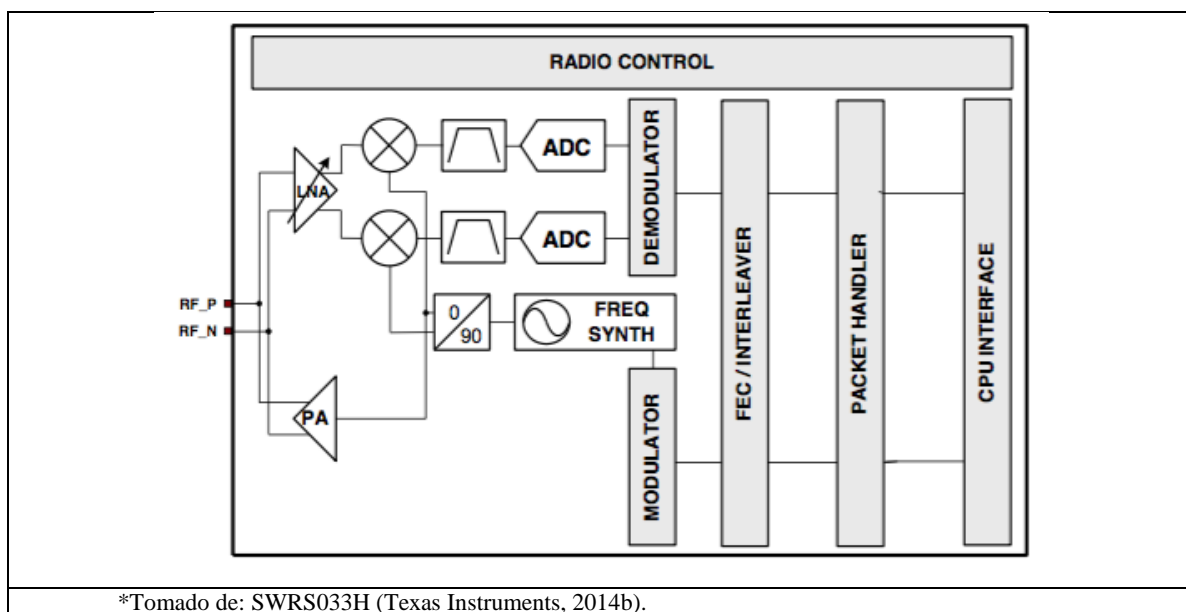


Figura 31: Diagrama de Bloques Módulo RF

Para el proceso de transmisión se incluye un sintetizador de frecuencia. Este sintetizador contiene un oscilador controlado por voltaje [VCO] y un desfasador de 90° para generar las señales de cuadratura [I/Q]. Un oscilador de cristal de 26/48 [MHz] es utilizado como referencia tanto para el sintetizador de frecuencia así como para los relojes

de operación de los módulos de conversión análoga digital (Texas Instruments, 2014b).

Las principales características de este módulo se resumen en la *Tabla 10*.

Tabla 10: Características Módulo RF

Característica	Descripción
Frecuencias de transmisión	<ul style="list-style-type: none"> • 300 – 348 [MHz] • 389 – 464 [MHz] • 779 – 928 [MHz]
Velocidad de transmisión de datos	<ul style="list-style-type: none"> • 0.8 – 500 [kBaud]
Uso del CPU	<ul style="list-style-type: none"> • Permite la utilización del módulo DMA para manejo de datos tanto en recepción como en transmisión
Modulación de datos	<ul style="list-style-type: none"> • FSK • GFSK • MSK • OOK • ASK
Sistemas de transmisión y recepción	<ul style="list-style-type: none"> • Sistema orientado al manejo de paquetes • Detección de direcciones y comunicación por paquetes de tamaño fijo y variable
Indicadores o sistemas adicionales	<ul style="list-style-type: none"> • Sistema de detección del uso del canal de transmisión para protocolos “listen-before-talk” • Implementación de un indicador del estado del enlace (LQI) • Implementación de un indicador digital de la fuerza en la señal recibida (RSSI)

Software de Recepción

El desarrollo de los algoritmos que se presentan a continuación tiene como sustento teórico un trabajo previo realizado por Texas Instruments. “CC USB Software Examples” es un manual de usuario en el que se describen dos ejemplos de programación para las tarjetas CC1111, CC2511 y CC2531 (Texas Instruments, 2009d). En uno de los ejemplos se implementa una configuración en el dispositivo USB para emular un puerto serial virtual (Virtual COM-port). En este manual los algoritmos implementados para la

comunicación inalámbrica se basan en el protocolo MRFI de Texas Instruments. Debido a que la transmisión de datos implementada en esta tesis se diseñó en base a una estructura y protocolo distinto, existe un proceso de adaptación en el código para que el receptor interprete correctamente los datos transmitidos.

Rutinas y algoritmos

El núcleo del módulo RF requiere de la programación de algunas rutinas y algoritmos para recibir los datos transmitidos inalámbricamente, analizar el contenido de cada paquete recibido y transmitir dicha información a la interfaz de usuario GUI a través del controlador USB. La programación de todas estas rutinas y subrutinas fue realizada en lenguaje C y assembler. A diferencia de las anteriores tarjetas utilizadas para la transmisión de datos, el dispositivo CC1111 USB Dongle utiliza el programa IAR Embedded Workbench IDE for 8051 para la programación. Este programa compila todas las rutinas y subrutinas de programación en un archivo [.hex]. Este archivo se utiliza después para eliminar cualquier información contenida en el procesador CC1111F32 utilizando el CC Debugger y el programa SmartRF Flash Programmer. Así mismo, el microprocesador se programa al cargar este archivo [.hex] en su memoria.

El ejemplo de programación utilizado para configurar la tarjeta USB como puerto serial [COM] se divide en varios archivos de cabecera y en varios archivos de rutinas y subrutinas (.c). Sin embargo, el lazo principal del sistema se encuentra dentro del archivo rf_modem.c, donde se encuentran declarados todos los procesos de inicialización y del funcionamiento y comportamiento de la tarjeta USB.

Archivo de lazo principal (rf_modem.c)

La rutina de lazo principal dentro del archivo `rf_modem.c` tiene dos funciones principales. Esta rutina, por un lado, configura e inicializa todos los registros asociados al controlador USB y al módulo RF y por otro lado, define el comportamiento y los procesos que se realizan de manera permanente en el microprocesador CC1111F32. En el diagrama que se presenta en la figura *Figura 32*, se detalla la rutina de inicialización y de lazo principal del sistema. Los procesos o subrutinas de mayor relevancia encontrados en este diagrama se detallan a continuación.

Comandos y procesos específicos del módulo USB

Como se puede observar en la *Figura 32*, existen varios procesos de configuración inicial y de funcionamiento continuo del módulo USB. Dentro de la configuración inicial se recalcan los siguientes puntos:

- La velocidad del cristal configurada es igual a 48 [MHz] con una frecuencia de referencia de 24 [MHz].
- La configuración de comunicación serial se realiza bajo los siguientes parámetros:
 - 38400 [bps]
 - 8 [bits] de datos
 - Paridad desactivada
 - 1 [bit] de parada
- El control de flujo se realiza mediante hardware [RTS/CTS]
- La bandera de control RTS se habilita cuando se completa la transacción de SETUP con la PC.

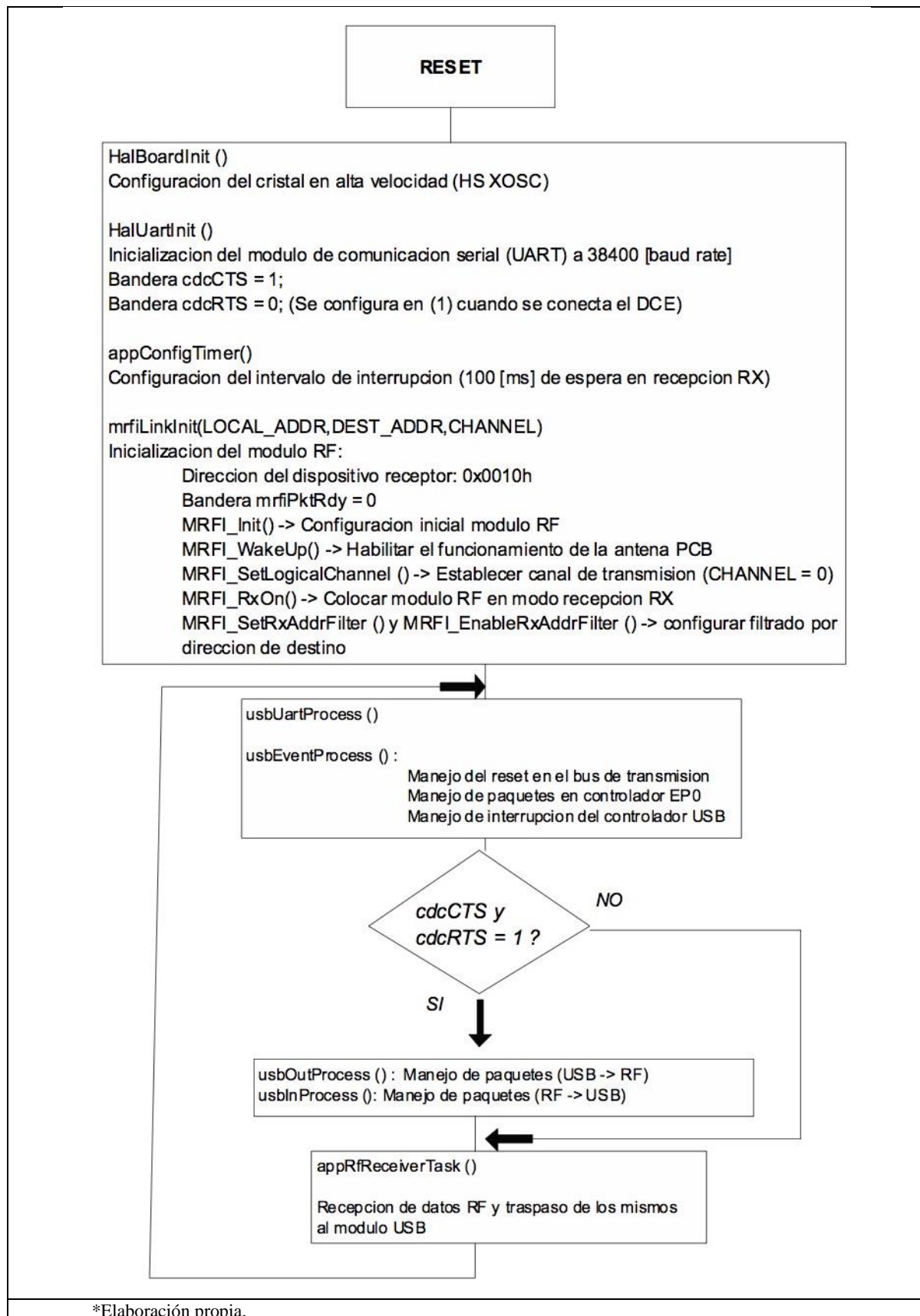


Figura 32: Flujoograma del lazo principal *rf_modem.c*

La comunicación entre los módulos RF y USB radica únicamente en la función `usbInProcess ()` debido a que el dispositivo USB es utilizado solo como receptor RF de datos y transmisor USB-Serial de los mismos. El detalle del funcionamiento de esta subrutina se presenta en el flujograma que se muestra en la *Figura 33*.

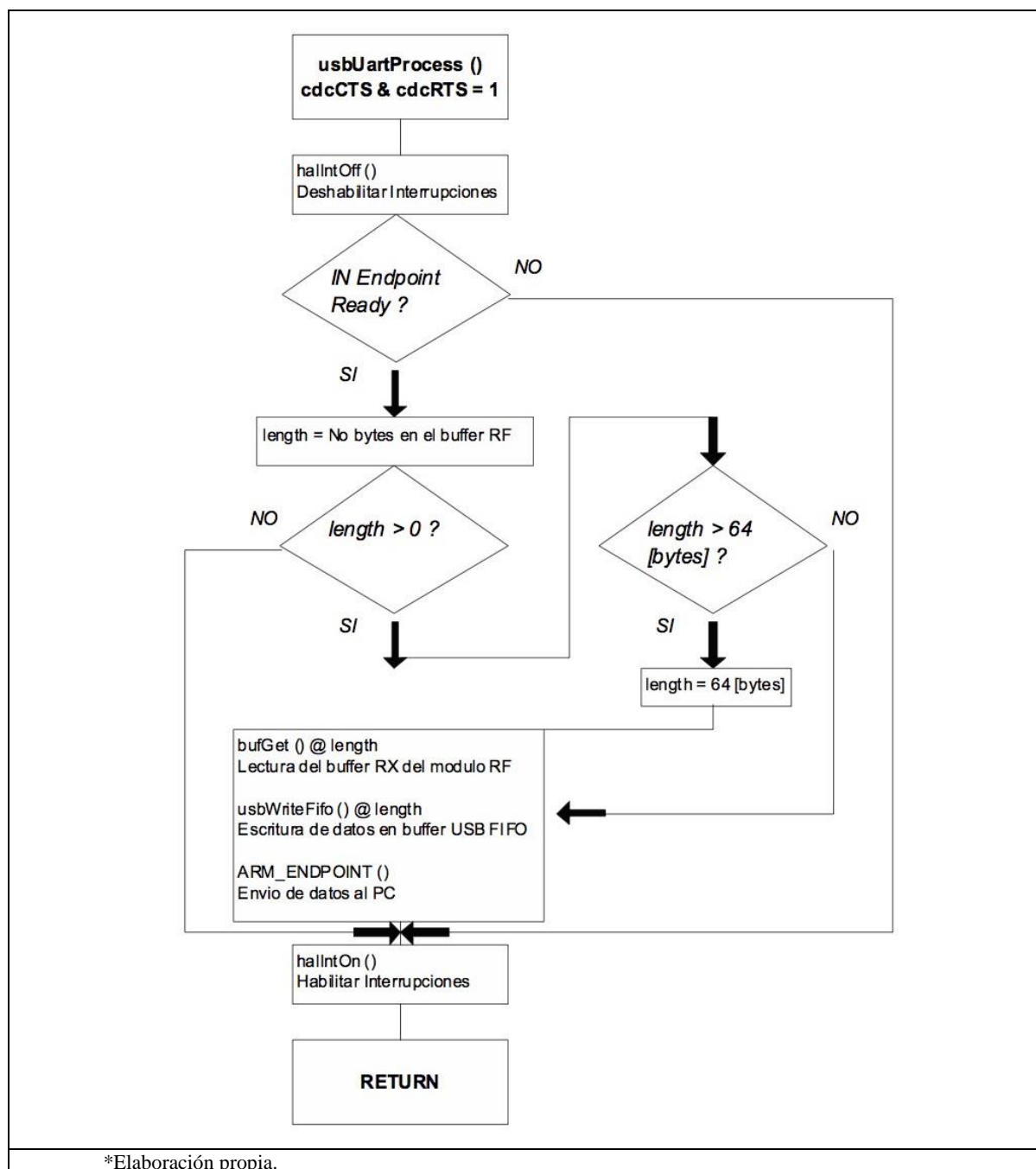


Figura 33: Flujograma USB IN

Comandos y procesos específicos del módulo RF

Para que la recepción inalámbrica de datos se correcta se necesitan también varios procesos y subrutinas de configuración. En la *Tabla 11* se resumen las configuraciones iniciales más importantes realizadas para este proceso en el módulo RF junto con sus respectivas descripciones. La mayoría de los valores utilizados en los registros de configuración son resultado de los parámetros de diseño utilizados en las tarjetas de transmisión y de los valores de configuración resultantes al utilizar el programa SmartRF Studio con los parámetros de diseño propuestos.

Tabla 11: Módulo RF C1111F32

Registros	Valores Asociados	Resultado
MCSM1	MCSM1[8:0] = 0X3C	<ul style="list-style-type: none"> Habilitación del sistema CCA: listen-before-talk. Mantenerse en modo RX después de una recepción
MCSM0	MCSM0[8:0] = 0X14	<ul style="list-style-type: none"> Autocalibración cuando se pasa del modo IDLE a RX o TX
PKTLEN	PACKET_LENGTH [8:0] = 0x1D	<ul style="list-style-type: none"> Tamaño del paquete FIFO RX o TX de 29 [bytes]
PKTCTRL0	PKTCRTL0[8:0] = 0x04	<ul style="list-style-type: none"> Cálculo del CRC en paquete de recepción [RX]
PKTCTRL1	PKTCRTL1[8:0] = 0x05	<ul style="list-style-type: none"> Bytes de RSSI y LQI en la cola de transmisión [TX] Filtrado por direccionamiento
FREQ2	FREQ[23:16] = 0x24	<ul style="list-style-type: none"> Unión de todas estas configuraciones FREQ[23:0] da como resultado una frecuencia de portadora de 867.99 [MHz]
FREQ1	FREQ[15:8] = 0x2A	
FREQ0	FREQ[7:0] = 0xAA	
PA_TABLE0	PA_TABLE0[8:0] = 0x50	<ul style="list-style-type: none"> Señal de salida con potencia de 0 [dBm] @ 868[MHz]
MDMCFG4	MDMCFG4[8:0] = 0xBA	<ul style="list-style-type: none"> Ancho de banda = 107.14 [KHz]
MDMCFG3	MDMCFG3[8:0] = 0xA3	<ul style="list-style-type: none"> Velocidad de transmisión = 38.36 [KBaud]
MDMCFG2	MDMCFG2[8:0] = 0x17	<ul style="list-style-type: none"> 30/32 – bits de sincronización Modulación GFSK
MDMCFG1	MDMCFG1[8:0] = 0x23	<ul style="list-style-type: none"> 4-bytes de preámbulo en modo de transmisión [TX]

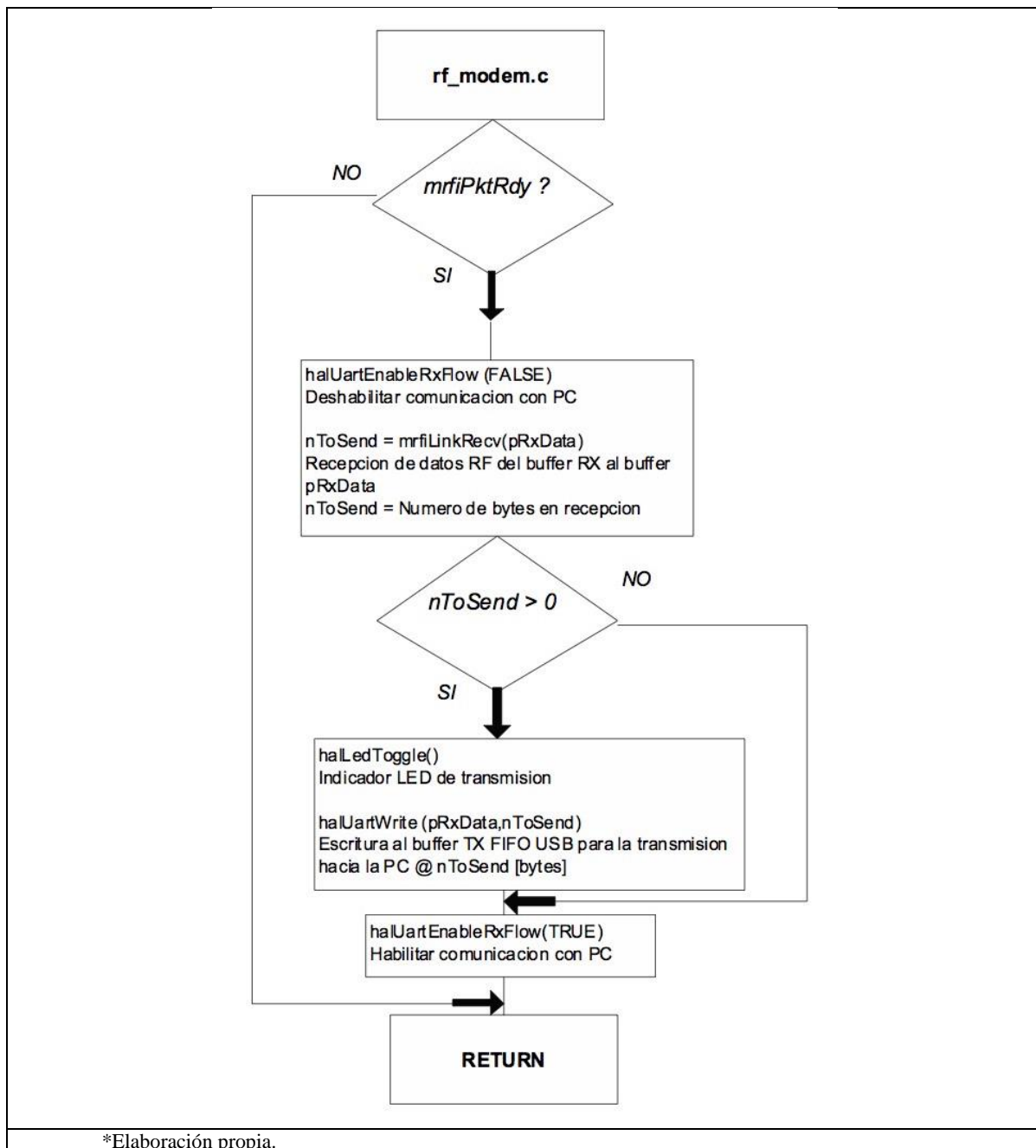


Figura 34: Flujo de la función `appReceiverTask()`

Para aclarar el funcionamiento continuo del módulo RF se presenta en la *Figura 34* el flujo de la función de recepción `appReceiverTask()`. Así mismo, en la *Figura 35* se presenta el flujo de la rutina de interrupción del módulo RF. Los diagramas presentados en esta sección completan la comprensión del funcionamiento receptor inalámbrico de datos, la transmisión y comunicación USB-Serial con la PC.

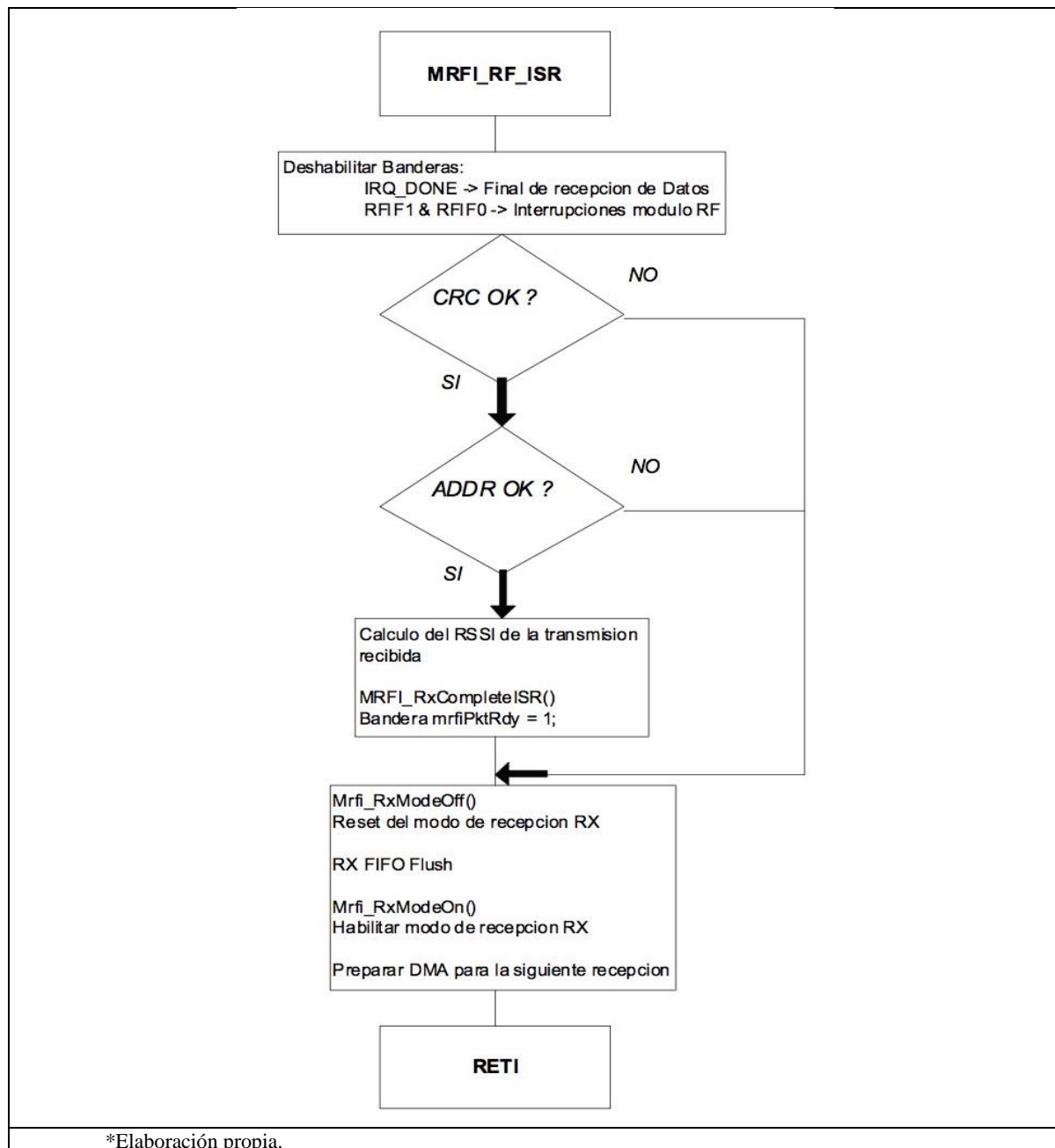


Figura 35: Flujograma CC1111 ISR

Interfaz Gráfica de Usuario [GUI]

Culminado el proceso de recepción y transmisión USB-Serial de los datos proporcionados por el sistema DAQ implementado, es necesaria la implementación de una interfaz gráfica que permita al usuario visualizar todas las variables de medición así como el estado de conexión de cada uno de los transmisores dentro de la red. Para la elaboración

de la GUI se utilizó el programa NI LabVIEW. Este programa es una plataforma de programación gráfica que permite el desarrollo de sistemas visuales que se integran con la mayoría de los dispositivos de comunicación.

Desencapsulamiento de la información

Como se menciona en los fundamentos de la transmisión inalámbrica, el rol del receptor es desencapsular la información contenida en los paquetes de transmisión. Cuando el receptor analiza los datos, del buffer de recepción se eliminan parámetros como sincronización y dirección de destino (Revisar el diagrama de la *Figura 26*). Por lo tanto, el parámetro que identifica la dirección de origen se convierte en el primero en ser analizado por la interfaz implementada. Posteriormente dicha interfaz examina todos los datos transmitidos hasta culminar el campo de datos (Data Field). Esta interfaz no analiza los bytes ubicados en la cola debido a que estos bytes son procesados únicamente por el receptor y no forman parte del buffer de transmisión USB-Serial. Es fundamental recordar que las variables de medición mayores a 8 [bits] de información son encapsuladas en paquetes secuenciales de 8 [bits]. En consecuencia, es necesario implementar en la GUI una función de concatenación de datos para volver a armar la variable separada y obtener así su valor original.

Implementación de la GUI utilizando NI LabVIEW

La interfaz gráfica desarrollada para el sistema DAQ implementado se presenta en la *Figura 36*. En este esquema, se puede observar que en el lado izquierdo se encuentran los campos de configuración para los parámetros de comunicación serial. Los valores por defecto se encuentran configurados con los valores utilizados en la programación del receptor. La excepción es el campo Serial Port que depende de la asignación COM

numérica de cada PC. En la parte inferior izquierda, se ubican los campos que activan o desactivan la exportación de los datos medidos a un archivo separado por comas [.csv].

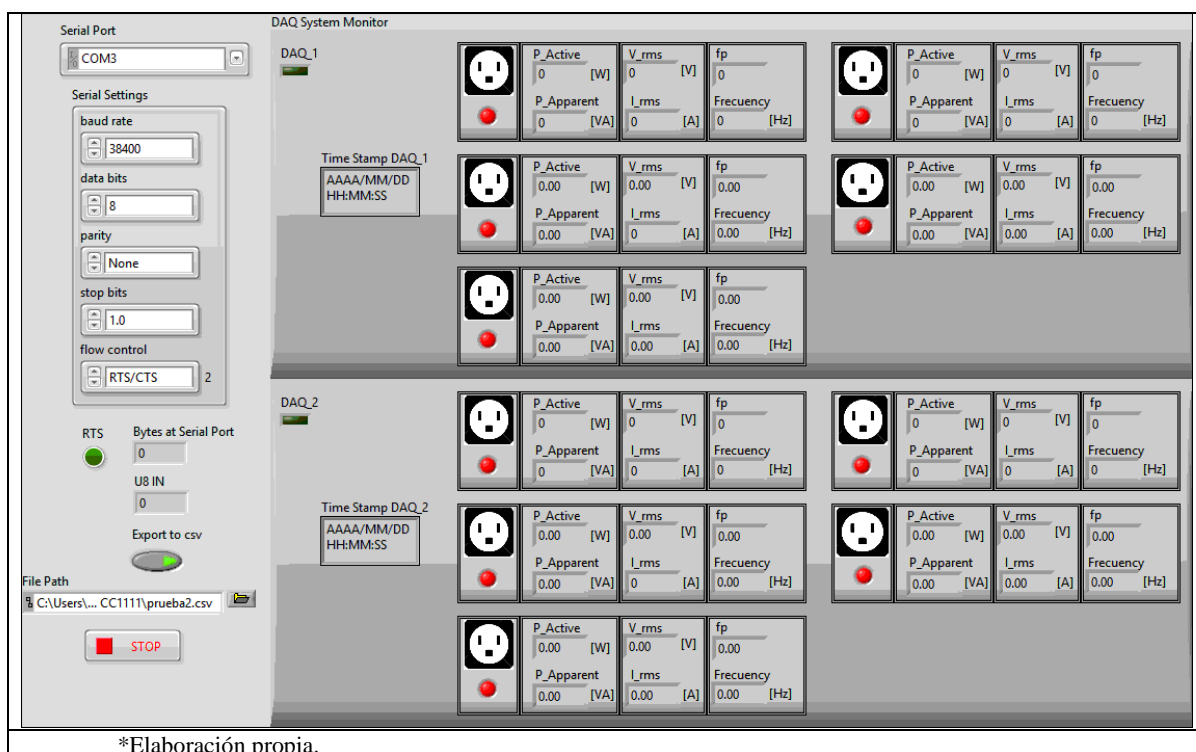


Figura 36: GUI

De igual forma, en esta sección se encuentran tres campos de resultados que identifican:

- RTS: Señala el estado de la bandera RTS del sistema receptor
- Bytes at Serial Port: Señala el número de bytes que se encuentran almacenados en el buffer de transmisión USB-Serial.
- U8 IN: Señala el valor entero numérico del primer byte recibido por la PC.

En la parte central y derecha de la interfaz, se encuentran graficados cada uno de los transmisores dentro de la red con sus cinco canales independientes de medición y su respectiva estampilla de tiempo. Cada canal, además de poseer su identificador de conectividad, muestra todas las variables de medición mencionadas anteriormente en

formato doble (DBL). Las características de la programación gráfica implementada en esta interfaz se resumen en la *Tabla 12* presentada a continuación.

Tabla 12: Características GUI

Característica	Descripción
Reseteo General del Sistema	Configuración de valores iniciales en todas las variables de medición y funcionamiento de la interfaz.
NI-VISA	Utilización de los diferentes módulos de comunicación serial NI-VISA.
Limites en la lectura	Establecimiento de un valor mínimo y un valor máximo para la lectura de datos.
Funcionamiento de tipo secuencial	Programación gráfica utilizando los módulos “Flat Sequence” y “Stacked Sequence” para controlar el orden de las operaciones.
Análisis FIFO de datos	Análisis de la cadena de bytes de información respetando el orden FIFO del buffer.
Sistema de actualización en indicadores de conectividad	Sistema continuo de actualización/reseteo en el LED o indicador de conexión.
Función de Concatenación en sub-módulo separado (var_calc.vi)	Programación gráfica del módulo de cálculo de variables de medición en una subrutina. Este sub-módulo utiliza un esquema de punto fijo para presentar las variables concatenadas separando la parte entera de la parte decimal codificada.
File I/O	Utilización de varios módulos para exportar los datos medidos a un archivo separado por comas [.csv] ubicado en el destino definido por el usuario [file path].

Capítulo 7: Resultados del Sistema de Adquisición de Datos

Estructura física resultante del sistema DAQ

Para este proyecto de tesis se manufacturaron dos tarjetas de adquisición de datos en base al diseño propuesto en el capítulo 3. En la *Figura 37* se presenta una imagen de la vista superior de una de las tarjetas físicas en comparación con su modelo, y en la *Figura 38* se presenta la misma comparación pero de la vista inferior. En estas imágenes se puede observar que la tarjeta física resultante se ajusta a la perfección con su diseño propuesto y que todos los componentes calzan dentro de sus respectivos patrones de soldadura [footprints]. Por esta razón, no fue necesario realizar ningún cambio a la estructura de la PCB.

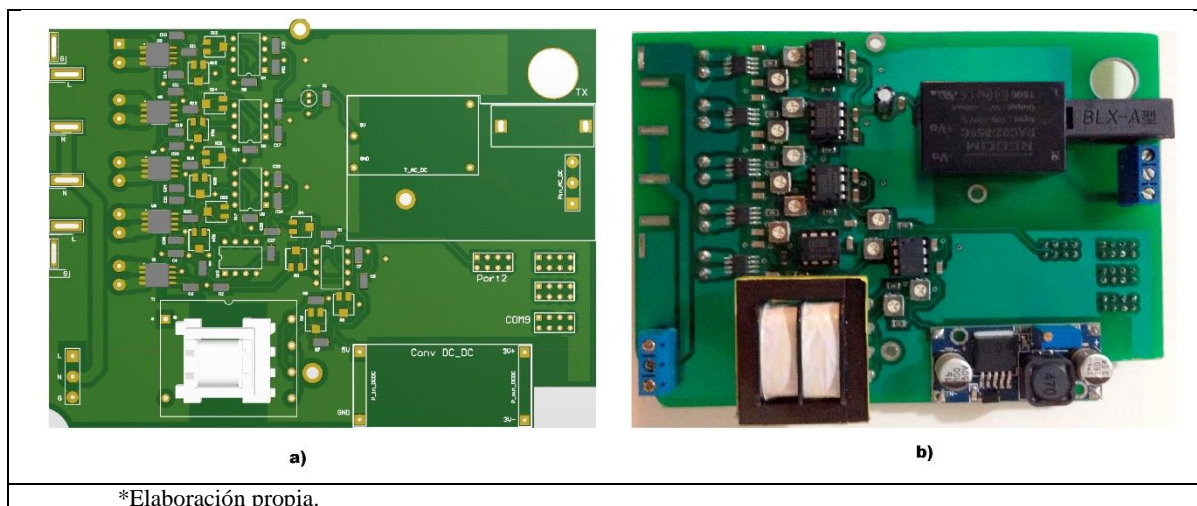


Figura 37: a) Modelo PCB b) Estructura PCB final (vista superior)

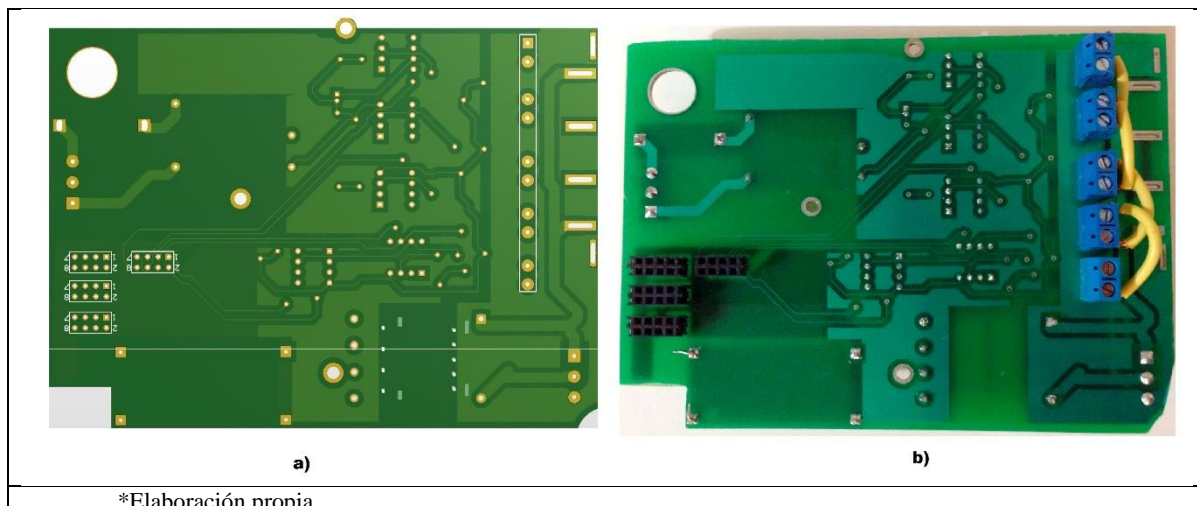


Figura 38: a) Modelo PCB b) Estructura PCB final (vista inferior)

Finalizada la elaboración de la tarjeta PCB, se manufacturó la estructura tipo regleta como también se propuso en el capítulo 3. En la *Figura 39* se presenta en la gráfica (a) el modelo tridimensional de la tarjeta PCB en conjunto con la tarjeta EM430F6137RF900 y en la gráfica (b) la foto de estas dos tarjetas conectadas en la realidad. Con estas gráficas se puede comprobar que el diseño físico de la PCB se ajusta correctamente a los puertos de salida de la tarjeta EM430F6137RF900.

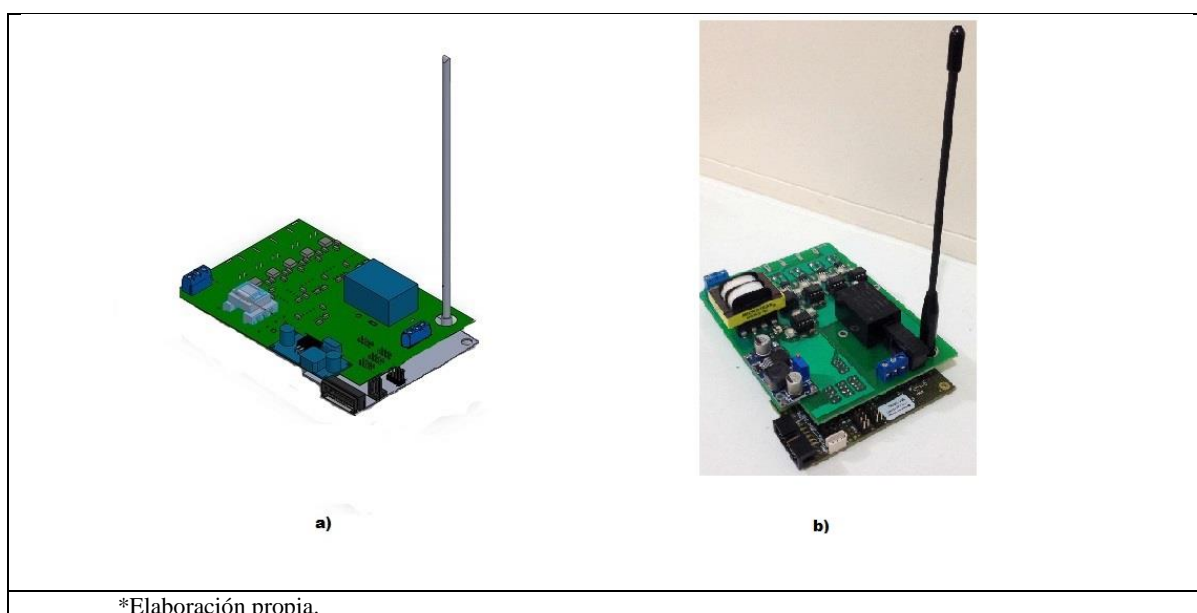


Figura 39: a) Modelo PCB + RF b) Estructura PCB + RF

Finalmente en la *Figura 40* se presenta en la gráfica (a) el modelo tridimensional de toda la estructura del sistema DAQ y en la gráfica (b) la foto de la estructura física real del mismo sistema.

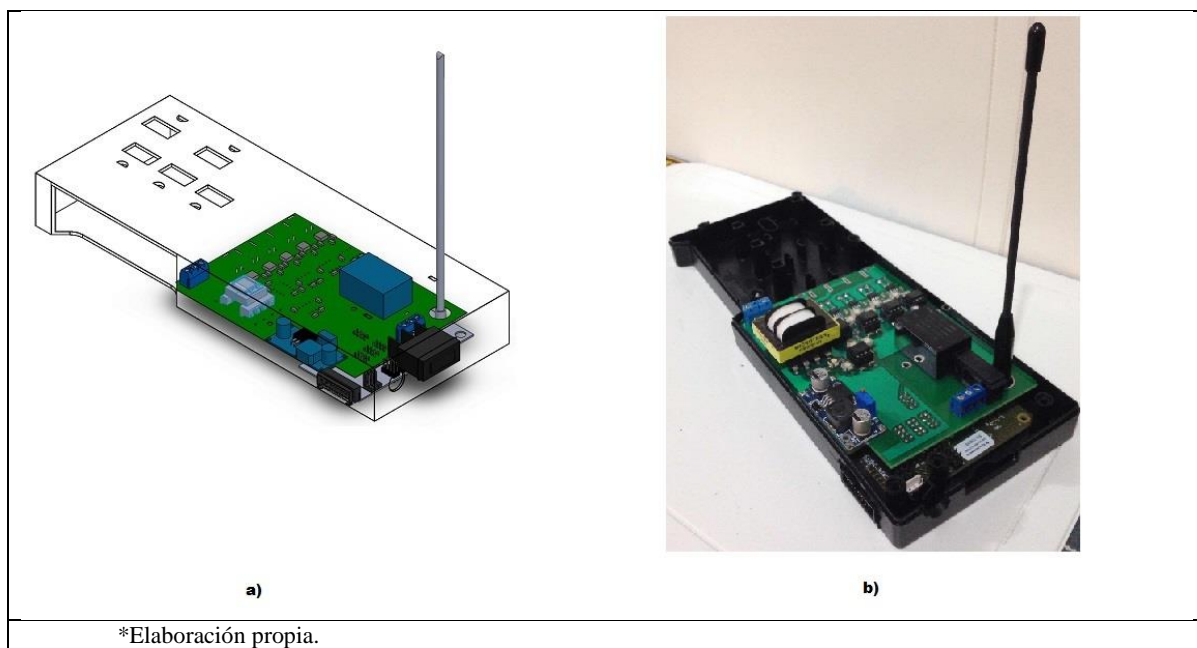


Figura 40: a) Modelo 3D Sistema DAQ b) Estructura Sistema DAQ

Resultados del funcionamiento del sistema DAQ implementado

Una vez finalizado el diseño e implementación de los diferentes procesos de medición, transmisión y recepción de las distintas variables que componen el sistema de medición, se procede a evaluar y analizar estadísticamente dichas variables con el objetivo de comprobar el correcto desempeño del sistema. Para realizar dicha evaluación, se procede a realizar la conexión descrita en la *Figura 41*. Como se puede observar, el sistema DAQ se conecta a una fuente de voltaje alterna tomada desde el tomacorriente. Esta fuente es la que se conecta al sensor de voltaje para su posterior medición. Para simular la variación de corriente en cada uno de los cinco sensores de medición se conecta una resistencia a una fuente de voltaje alterna controlada [VARIAC]. Dicha carga

[resistencia] se conecta a los cinco sensores de corriente para poder así observar el funcionamiento de todo el sistema de forma simultánea. Es importante mencionar que se conecta un voltímetro en paralelo con la alimentación del sistema DAQ y un amperímetro en serie con la carga con el objetivo de tener una referencia real instantánea de los valores medidos.

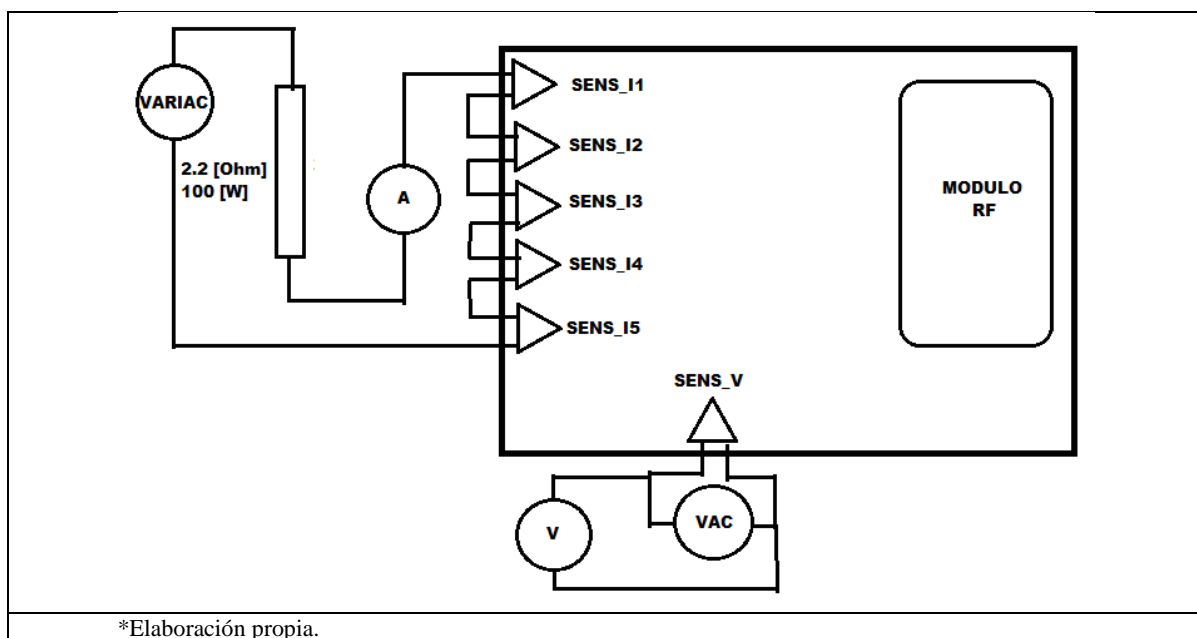


Figura 41: Diagrama Conexión de Prueba DAQ

Para el análisis de los resultados se tomaron 15 diferentes sets de datos, cada uno correspondiente a un valor diferente de corriente de alimentación. Cada valor de alimentación se mantuvo constante por un promedio de 3 a 4 minutos con el objetivo de asegurar una recepción de paquetes de información mínima por cada set. Cada set de datos incluye un valor real de corriente medido por el amperímetro, un valor real de voltaje medido por el voltímetro, y todos los valores medidos por el sistema DAQ referentes a la alimentación provista y expuestos en el archivo de exportación [.csv] proveniente de la interfaz GUI previamente desarrollada. El número de mediciones receptadas por set es

dependiente de la tasa de transmisión entre el transmisor y el receptor y por lo tanto, puede variar entre los diferentes sets.

Al finalizar el período de toma de datos del sistema DAQ, se separa y se clasifica la información por cada uno de los canales de medición. Debido a la imposibilidad de obtener valores reales constantes de voltaje y corriente en el período de medición, los valores reales registrados reflejan un promedio empírico respectivo. Por esta razón y para estandarizar el tamaño entre los sets, se comparan los promedios de todos los datos medidos por el sistema DAQ con los datos reales de voltaje y corriente. Adicionalmente, se compara la potencia activa medida con un valor real de potencia activa resultante de la multiplicación entre la medición real de corriente con la de voltaje. Esta operación se puede realizar debido a que los valores reales medidos reflejan valores efectivos [rms] y se asume una forma sinusoidal para estas dos mediciones.

Tabla 13: Mediciones Canal_1 (promedios)

V_real_1 [V]	V_rms_1 [V]	I_real_1 [A]	I_rms_1 [A]	P_real_1 [W]	P_Active_1 [W]	Factor_P_1	Frecuency_1 [Hz]
114.90	113.47	0.39	0.77	44.81	60.17	0.68	60.02
113.10	112.36	0.74	1.30	83.69	129.69	0.88	59.96
113.10	112.10	1.13	1.95	127.80	196.04	0.89	60.00
113.40	112.56	1.38	2.36	156.49	239.05	0.89	59.98
113.10	112.19	1.65	2.82	186.62	284.50	0.89	59.99
114.90	113.25	2.07	3.49	237.84	357.40	0.89	59.99
113.70	112.58	2.47	4.17	280.84	423.45	0.89	60.00
113.90	113.11	2.87	4.85	326.89	493.53	0.89	60.02
113.70	113.22	3.16	5.34	359.29	541.12	0.89	60.01
114.60	113.48	3.42	5.75	391.93	586.42	0.89	60.01
114.80	113.41	3.85	6.43	441.98	654.92	0.89	60.00
113.80	112.92	4.22	7.14	480.24	727.46	0.89	59.96
112.90	111.82	4.79	8.08	540.79	816.78	0.90	59.98
113.40	112.86	5.15	8.63	584.01	874.15	0.90	60.00
115.80	114.62	5.65	9.49	654.27	976.37	0.89	59.99

Toda esta información se presenta en tablas separadas por canal de medición. En la *Tabla 13* se presenta los datos obtenidos por el canal 1. En el anexo F1.1 se presentan tablas similares para los otros canales de medición y el análisis de las mismas se desarrolla en las siguientes secciones. Cabe recalcar que las mediciones de potencia aparente no se incluyen en ninguna tabla de promedios ya que estos valores son el resultado de la multiplicación entre el voltaje efectivo y la corriente efectiva. Por lo tanto, al analizar estas dos mediciones por separado se analiza indirectamente los resultados de la potencia aparente.

Análisis del funcionamiento del sistema DAQ implementado

Como se puede observar en la *Tabla 13*, existe una evidente variación entre los datos medidos con respecto a los datos reales. En las mediciones de voltaje existe una distancia promedio de 1 [V] entre los datos reales y los datos medidos. Sin embargo, en las mediciones de corriente esta distancia es mayor y se incrementa a medida que aumenta la corriente de alimentación [dato real]. A su vez, esto tiene su efecto sobre las mediciones de potencia activa ya que este valor es el reflejo de la multiplicación entre la corriente y el voltaje, así como también sobre el factor de potencia que se encuentra notoriamente por debajo de su valor teórico de 1 [para cargas totalmente resistivas]. Finalmente, en la última columna se puede observar las mediciones de frecuencia de la onda de voltaje de alimentación, todas cercanas a su valor teórico de 60 [Hz]. Para corroborar estadísticamente el análisis previamente descrito, se realizaron algunas pruebas estadísticas sobre los datos expuestos utilizando el software Minitab 17. Tanto para el voltaje como para la corriente y la potencia se realizaron pruebas t-pareadas para determinar si los valores medidos son significativamente diferentes a los valores reales. En cambio, para el

factor de potencia y para la frecuencia se realizaron pruebas t-de una muestra para determinar si los valores medidos son iguales a sus valores teóricos. La hipótesis nula de que los valores reales son iguales a los valores medidos se rechaza para todas las pruebas t-pareadas; esto quiere decir que existe una diferencia significativa entre estos dos grupos de datos. Como era de esperarse, la prueba t-de una muestra para el factor de potencia arroja que estos datos son significativamente distintos de uno. Sin embargo, la hipótesis nula en la prueba t-de una muestra para la frecuencia no se rechazó, lo que indica que existe un 95% de confiabilidad de que ésta es igual a 60 [Hz]. Un resumen de todas estas pruebas y sus resultados se muestran en el anexo F1.2

Factores estadísticos de corrección

A partir del análisis expuesto en la sección anterior, se puede evidenciar una falla en el desempeño del sistema de medición. Es importante mencionar que existen algunas características tanto físicas como digitales que insertan un desfase reflejado en el dato final de medición. Dentro de las características físicas, errores en la precisión de calibración y la inexactitud o rango de tolerancia de los componentes utilizados alteran el comportamiento analógico de las señales de entrada. Dentro de las características digitales, la eliminación de bits de información que codifican la parte decimal de las variables de medición y el desfase en el tiempo de muestreo por canal disminuyen también la precisión en los datos medidos finales. En consecuencia, todos estos errores se acumulan de la misma manera en cada canal de medición, por lo que se espera que estos errores sigan una tendencia o patrón.

Para determinar si los errores en las mediciones siguen efectivamente un patrón, se realizaron regresiones lineales para las mediciones de voltaje, corriente y potencia activa.

Los resultados de cada regresión se presentan en una gráfica de dispersión junto con su línea de tendencia y su estadístico R^2 asociado. Para el factor de potencia no se realizó ningún análisis adicional debido a su relación implícita con los análisis de potencia activa y potencia aparente. Así mismo, para las mediciones de frecuencia no se realizaron análisis posteriores debido a que se demostró que son estadísticamente iguales a su valor teórico. En la *Figura 42*, *Figura 43* y *Figura 44*, se muestran las gráficas de dispersión para el voltaje, corriente y potencia activa respectivamente del canal uno. En el anexo F1.3 se muestran las gráficas de dispersión restantes.

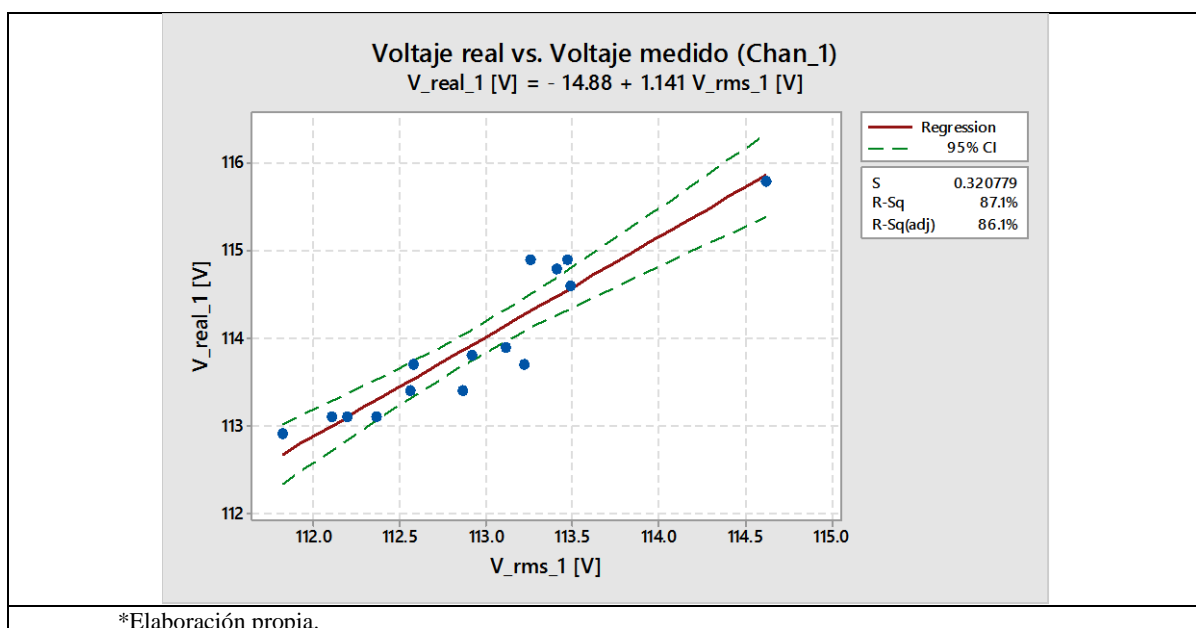


Figura 42: Gráfica de dispersión voltaje Chan_1

Como se puede observar en la *Figura 43* y *Figura 44*, los datos medidos tanto de corriente como potencia activa tienen una relación lineal fuerte con los datos reales. Esto se puede notar visualmente al ver que todos los puntos caen sobre la recta de tendencia y también al observar su estadístico $R^2 = 100\%$. No obstante, como se puede observar en la *Figura 42* la gráfica de dispersión para el voltaje no muestra una conclusión tan robusta debido a que no todos los puntos se encuentran dentro del intervalo de confianza. Esto se

refleja a su vez en un valor inferior del estadístico R^2 . A pesar de que a simple vista este valor parezca no significativo, una prueba de significancia del R de Pearson muestra que sí lo es. Resultados similares se encontraron para cada uno de los otros cuatro canales.

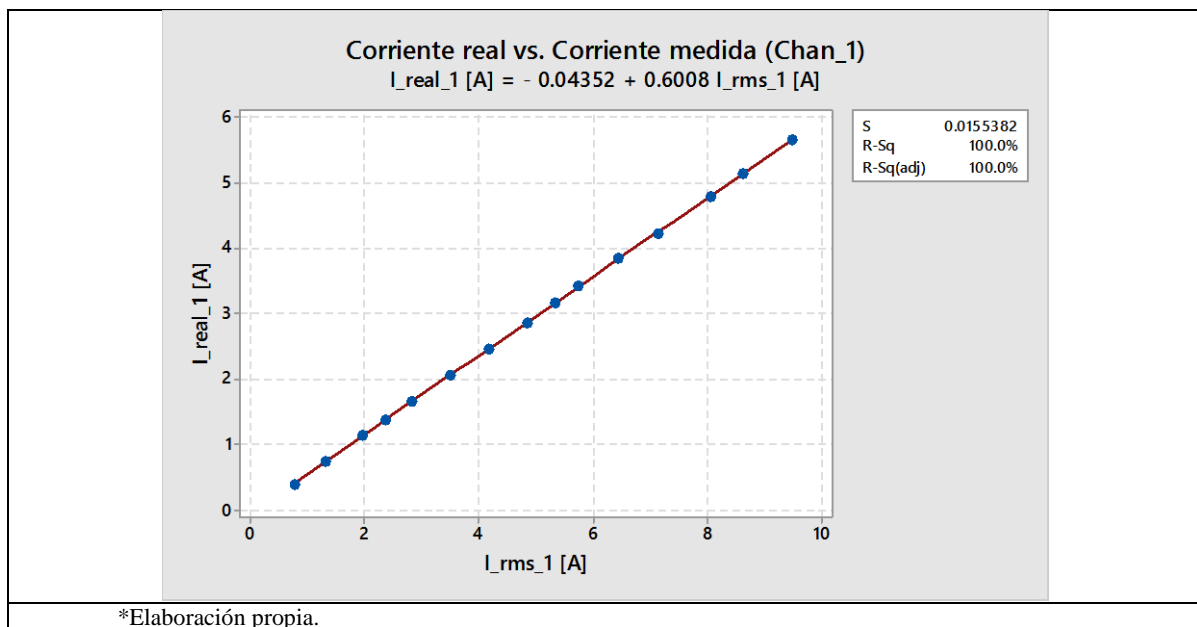


Figura 43: Gráfica de dispersión corriente Chan_1

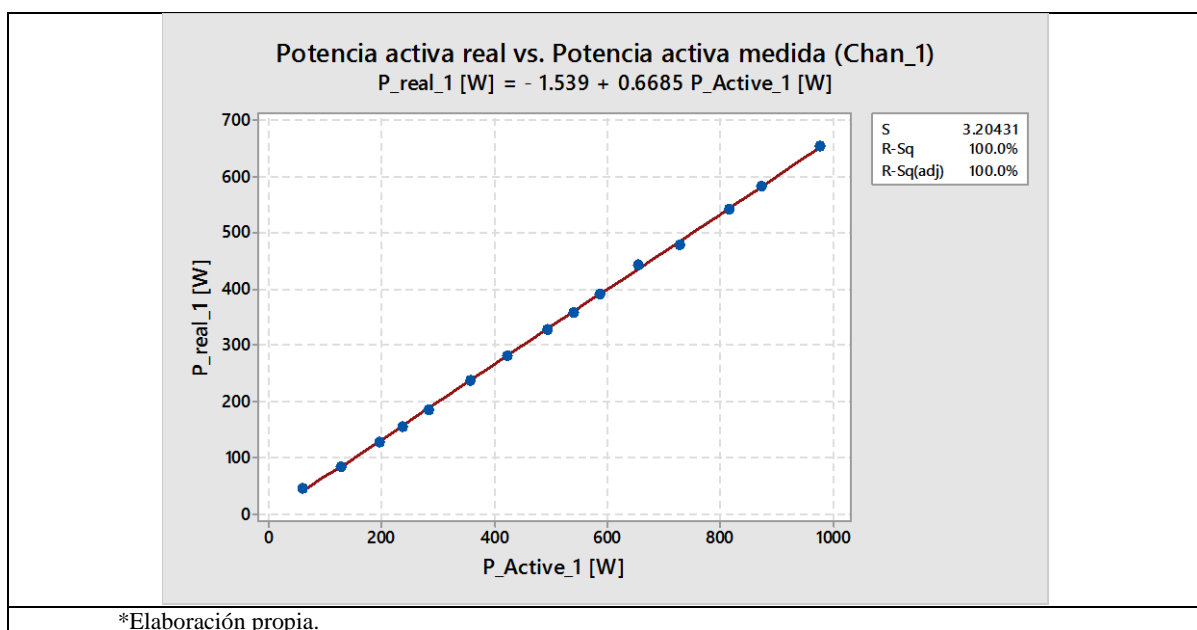


Figura 44: Gráfica de dispersión potencia activa Chan_1

Resultados del funcionamiento del sistema DAQ con funciones de corrección

Una vez caracterizados los errores en las mediciones del sistema DAQ implementado, se propone una metodología para reducir los mismos. Esta metodología utiliza las ecuaciones de regresión presentadas en los gráficos de dispersión. Estas ecuaciones se programan como una función de corrección para los datos medidos dentro de cada subrutina de cálculo en el archivo foreground.c. Modificado el código de medición, se toman nuevos sets de información utilizando la misma metodología presentada en la sección anterior. Al finalizar la toma de datos, toda la información recolectada se separa nuevamente en tablas por canal de medición. En la *Tabla 14* se presenta los nuevos datos obtenidos del canal 1 y en el anexo F1.4 se presentan tablas similares para los otros canales de medición.

Tabla 14: Mediciones Canal 1 con corrección (promedios)

V_real_1 [V]	V_rms_1 [V]	I_real_1 [A]	I_rms_1 [A]	P_real_1 [W]	P_Active_1 [W]	Factor_P_1	Frecuency_1 [Hz]
114.300	114.496	0.000	0.145	0.000	0.000	0.195	59.989
115.800	114.628	0.400	0.415	46.320	44.859	0.873	59.991
117.200	118.074	0.920	0.925	107.824	105.746	0.937	60.019
115.200	114.632	1.130	1.121	130.176	126.022	0.952	59.965
114.100	114.285	1.630	1.622	185.983	182.140	0.960	59.988
114.000	114.042	2.250	2.258	256.500	253.744	0.966	60.011
113.300	113.566	2.610	2.610	295.713	290.873	0.970	59.971
113.100	113.445	2.830	2.840	320.073	316.032	0.971	60.011
116.300	115.834	3.120	3.115	362.856	352.996	0.965	59.979
115.900	115.827	3.500	3.487	406.350	395.041	0.966	60.017
114.900	114.505	4.120	4.075	473.388	458.114	0.971	59.990
114.500	114.515	4.450	4.391	509.525	495.373	0.971	59.987
115.100	115.377	4.710	4.710	542.121	532.151	0.970	59.999
115.600	115.292	5.360	5.366	619.616	609.697	0.970	59.991
115.800	115.876	5.780	5.723	669.324	647.742	0.972	59.997

Análisis del funcionamiento del sistema DAQ con funciones de corrección

Como se puede observar en los resultados expuestos en la *Tabla 14*, las mediciones utilizando las funciones de corrección son más precisas. Por esta razón, para evaluar el desempeño del sistema, ya no se realizan pruebas t-pareadas o t-de una muestra sino que se calcula simplemente el error porcentual entre las mediciones reales y las obtenidas por el sistema DAQ. La forma de calcular este error se describe en la ecuación (43).

$$error (\%) = \frac{Dato_{real} - Dato_{medido}}{Dato_{real}} \times 100 \quad (43)$$

En la *Tabla 15* se muestran los errores porcentuales de las mediciones en el canal 1 calculados a partir de los datos en la *Tabla 14*. Adicionalmente en esta tabla se incluye un promedio del error en la medición.

Tabla 15: Tabla de error Canal 1

Set de datos	Error Voltaje	Error Corriente	Error P. Activa	Factor de Potencia
1	1.01%	3.65%	3.15%	12.73%
2	0.75%	0.59%	1.93%	6.28%
3	0.49%	0.76%	3.19%	4.81%
4	0.16%	0.46%	2.07%	4.00%
5	0.04%	0.37%	1.07%	3.44%
6	0.23%	0.01%	1.64%	3.01%
7	0.31%	0.34%	1.26%	2.93%
8	0.40%	0.16%	2.72%	3.51%
9	0.06%	0.37%	2.78%	3.40%
10	0.34%	1.09%	3.23%	2.93%
11	0.01%	1.33%	2.78%	2.94%
12	0.24%	0.00%	1.84%	2.96%
13	0.27%	0.11%	1.60%	3.02%
14	0.07%	0.99%	3.22%	2.85%
Error medio	0.31%	0.73%	2.32%	4.20%

Como se puede observar en este canal, el error medio es menor al 1% para las mediciones de voltaje y corriente lo que indica una acertada corrección de los datos. Sin embargo, las mediciones de potencia y factor de potencia presentan un error mayor pese a que estos no superan al 5%. Este aumento se puede justificar con los errores para voltaje y corriente ya que las mediciones de potencia y factor de potencia son dependientes de los mismos. En el anexo F1.5 se muestran las tablas de errores para los otros cuatro canales en las cuales los resultados que se pueden encontrar son similares a los resultados del canal 1. Por lo tanto, se puede afirmar que las mediciones realizadas por el sistema DAQ tienen un alto nivel de precisión para cargas resistivas.

Capítulo 8:

Conclusiones y Recomendaciones

Conclusiones

Culminado el proceso de implementación y evaluación del sistema de medición propuesto, se puede concluir que existen algunos elementos que son fundamentales para un correcto dimensionamiento de los circuitos. La aplicación de la teoría para predecir el comportamiento del circuito y la simulación del mismo para corroborar dicha predicción son factores claves en el éxito del diseño. Para la elaboración de la tarjeta PCB es necesario respetar algunos parámetros eléctricos. Por ejemplo, los caminos de conducción se implementaron en un sistema multicapa [2 capas], lo cual permitió mejorar la distribución de los mismos. Específicamente se realizó en esta tarjeta una distinción entre los caminos de potencia y los caminos de señales digitales: para los caminos de potencia se utilizó un tamaño estándar de 2.5 [mm] de ancho mientras que para los digitales se utilizó un tamaño estándar de 0.25 [mm] de grosor. Así mismo, se puede concluir que la separación de las tierras entre estos dos grupos de señales [digitales y potencia] cumple un rol fundamental en la protección del circuito, la estabilidad de medición y la estandarización de un voltaje de referencia.

Con respecto a la estructura lógica o digital del sistema DAQ, se puede concluir que para poder comprender la programación realizada por trabajos previos, adaptarla a las nuevas necesidades del sistema propuesto y programar nuevas rutinas o subrutinas de procesamiento es totalmente necesario un conocimiento previo de los lenguajes de programación utilizados [C y Assembler] en la representación hexadecimal de la

información binaria así como un conocimiento en el manejo de registros de información y rutinas de interrupción por parte de los microprocesadores. De igual forma, se puede concluir que para el procesamiento de variables que ocupen gran tamaño de información, es imprescindible la reducción de los bits menos significativos a pesar de que esta metodología introduzca un error en la medición por pérdida de información decimal. Para finalizar, se puede concluir que la utilización de una aritmética de punto fijo para variables binarias facilita las operaciones matemáticas y la codificación de sus resultados en números con parte entera y parte decimal.

Con respecto a la transmisión inalámbrica, se puede concluir que se debe definir claramente los protocolos de comunicación utilizados tanto en el receptor como en el o los transmisores para establecer una comunicación exitosa. En primer lugar, establecer un protocolo de encapsulamiento propio tiene sus ventajas así como sus desventajas: un protocolo propio se estructura en base a las necesidades del programador sin embargo, no se adapta a ningún protocolo estándar por lo que no sería compatible con ningún otro dispositivo transceptor. En segundo lugar, la implementación de un protocolo de enrutamiento sin confirmación en la recepción [ACK] aumenta la velocidad de transmisión pero no asegura una completa recepción de todos los paquetes enviados. Para finalizar, se puede concluir que el protocolo de mayor importancia dentro de la red es el protocolo de acceso al medio [MAC] ya que éste evita la saturación del medio (aire) cuando permite el uso del medio solo al dispositivo que tiene el “token” de transmisión. Esto a su vez es lo que permite una correcta recepción de los paquetes enviados pues de no existir este mecanismo, el receptor solo recibiría la información de la tarjeta más próxima.

Con respecto a la interfaz gráfica de usuario [GUI], se puede concluir que el uso del programa LabVIEW facilita en muchos aspectos el procesamiento e interpretación de los datos receptados, y con ello la creación de una plataforma virtual que muestra al usuario final los valores medidos por el sistema DAQ. Dentro de los aspectos de procesamiento matemático, LabVIEW permite el uso de módulos que manejan matemáticamente la información receptada. Asimismo, LabVIEW presenta módulos de punto fijo para el tratamiento de información binaria codificada en su parte real y decimal. Dentro de los aspectos de comunicación, LabVIEW facilita la comunicación serial entre la PC y el dispositivo USB receptor. Por último, LabVIEW optimiza el proceso de análisis estadístico ya tiene habilitada la opción de exportación de datos que pueden ser utilizados por otros programas como Microsoft Excel o Minitab.

Finalmente, con respecto a la evaluación del desempeño del sistema DAQ implementado, se puede afirmar que los primeros datos medidos [sin funciones de corrección] contienen un elevado porcentaje de error ocasionado tanto física como digitalmente. Este error que fue determinado como estadísticamente significativo, se puede observar en la diferencia que existe entre los datos reales y los datos medidos. Sin embargo, un análisis matemático permitió reducir estos errores a través de la implementación de funciones de corrección basadas en las ecuaciones de una regresión lineal. Una vez implementadas estas funciones de corrección, se observa que las mediciones se ajustan a los valores reales y que los errores de cálculo se reducen significativamente, lo cual permite concluir de manera general un correcto desempeño del sistema DAQ.

Recomendaciones

En base al trabajo realizado, a los resultados obtenidos y a la experiencia adquirida se recomiendan algunas modificaciones físicas para optimizar y mejorar el funcionamiento del sistema DAQ implementado. Con respecto a la tarjeta PCB se sugiere:

- Utilizar potenciómetros solo en los circuitos en los que el valor de resistencia es crítico en el comportamiento de la salida del mismo para disminuir el error por calibración.
- Colocar puntos de prueba para facilitar el acceso de las puntas de prueba de los multímetros para comprobar el funcionamiento del circuito y permitir una mejor calibración del mismo.
- Utilizar en lo posible valores de resistencias comunes o exactas.

Así mismo, para la estructura física del sistema se recomienda diseñar una nueva carcasa física específicamente a partir del modelo de la tarjeta PCB.

Adicionalmente, se sugieren las siguientes modificaciones en el sistema de comunicación:

- Utilizar un protocolo de encapsulamiento y comunicación estándar para permitir la compatibilidad con otros sistemas inalámbricos.
- Utilizar un protocolo de comunicación basado en la confirmación en la recepción [ACK] para asegurar la completa transmisión de datos.
- Implementar una comunicación bilateral entre PC y el receptor RF y del receptor RF a los transmisores RF para modificar datos de los transmisores a través de la GUI.

Finalmente, para la interfaz GUI se recomienda utilizar directamente el módulo de exportación a Excel de LabVIEW y eliminar la exportación de datos a un archivo [.csv].

Referencias Bibliográficas

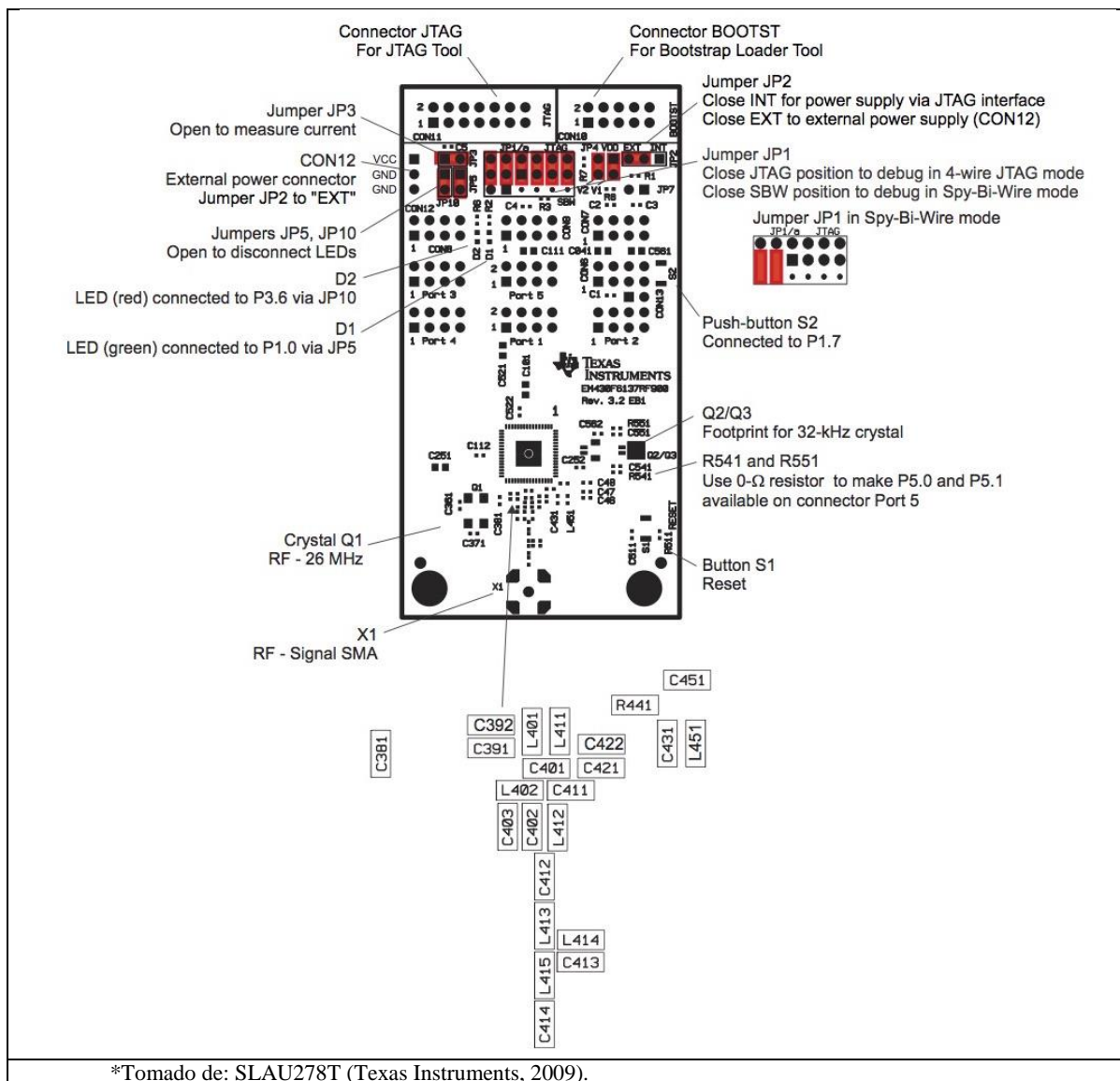
- Allegro Microsystems. (2012). *ACS712 Datasheet*. Massachusetts: Allegro Microsystems.
- Buratti, C., Conti, A., Dardari, D. & Verdone, R. (2009). An Overview on Wireless Sensor Networks Technology and Evolution. *Sensors*, 9, 6870–6896. doi: 10.3390/s90906869
- Cisco Systems. (2012). *Token Ring/IEEE 802.5*. Recuperado el 4 de junio de 2015 de: http://docwiki.cisco.com/wiki/Token_Ring/IEEE_802.5
- Cisco Systems. (2014). *Internetwork Basics*. Recuperado el 4 de junio de 2015 de http://docwiki.cisco.com/wiki/Internetworking_Basics#What_Is_an_Internetwork.3F
- Deitel, P. & Deitel, H. (2009). *Cómo Programar* (6ª ed.). México D.F: Prentice Hall.
- Dorf, R. & Svodoba, J. (2011). *Circuitos Eléctricos* (8ª ed.). México D.F: Alfaomega.
- Fitzgerald, F., Kingsley, C. & Umans, S. (2004). *Máquinas Eléctricas* (6ª ed.). México D.F: McGraw Hill.
- Forstner, P. (1999). *FIFO Architecture, Functions, and Applications*. Dallas: Texas Instruments Incorporated.
- Gram-Hanssen, K. (2014). New needs for better understanding of household's energy consumption – behaviour, lifestyle or practices?. *Architectural Engineering & Design Management*, 10(1/2), 91-107. doi:10.1080/17452007.2013.837251
- Hać, A. (2003). *Wireless Sensor Network Designs*. Honolulu: Wiley.

- Harris, D. & Harris, S. (2013). *Digital Design and Computer Architecture* (2^a ed.). Waltham: Elsevier.
- Holmberg, D. (2011). Demand Response And Standards. *ASHRAE Journal*, 53(11), B23-B28.
- Ingle, V. & Proakis, J. (2012). *Digital Signal Processing* (3^a ed.). Stamford: Cengage.
- Kiliccote, S., Piette, M. & Ghatikar, G. (2011). Smart Buildings and Demand Response. *AIP Conference Proceedings*, 1401(1), 328-338. doi:10.1063/1.3653861
- Lasheng, Y., Jie, L. & Renjie, L. (2013). An effective data collection algorithm for wireless sensor network. *Computing*, 95(9), 723-738. doi:10.1007/s00607-012-0249-1
- Lin, Z., Jianming, L., Hongzhou, L. & Zhiyong, P. (2014). Study on the Effectiveness of Wireless Ad hoc Networks Routing Protocols. *Information Technology Journal*, 13(7), 1356-1360. doi:10.3923/itj.2014.1356.1360
- Mesganaw, M. (2012). *Energy Meter Code Library for 1-Phase to 3-Phase Using MSP430 Family*. Dallas: Texas Instruments Incorporated.
- Mishra, S. & Kaur, P. (2014). Energy Efficient Neighbour Selection For Flat Wireless Sensor Networks. *International Journal Of Computer Science & Information Technology*, 6(4), 95-102. doi:10.5121/csit.2014.4511
- Morris, A. & Langari, R. (2012). *Measurement and Instrumentation Theory and Application*. San Diego: Elsevier.
- Namtvedt, S. (2009). *Design Note DN500*. Dallas: Texas Instruments Incorporated.

- Nokhanji, N. & Hanapi, Z. (2014). A Survey on Cluster-based Routing Protocols in Wireless Sensor Networks. *Journal Of Applied Sciences*, 14(18), 2011-2022. doi:10.3923/jas.2014.2011.2022
- Singh Bhadoria, R., Singh Tomar, G. & Kang, S. (2014). Proficient Energy Consumption Aware Model in Wireless Sensor Network. *International Journal Of Multimedia & Ubiquitous Engineering*, 9(5), 27-36. doi:10.14257/ijmue.2014.9.5.03
- Sohraby, K., Minoli, D. & Znati, Z. (2007). *Wireless Sensor Networks*. New Jersey: Wiley.
- Stankovic, J. (2006). *Wireless Sensor Networks*. Artículo no publicado. Facultad de Ciencias Computacionales, Universidad de Virginia, Charlottesville, Virginia.
- Svehla, K. (2011). *A Specification for Measuring Domestic Energy Demand Profiles* (Tesis de maestría no publicada). Reino Unido: Universidad de Strathclyde.
- Tam, K. (2001). *Current-Transformer Phase-Shift Compensation and Calibration*. Dallas: Texas Instruments Incorporated.
- Texas Instruments. (1998). *CC430 Family User Guide*. Dallas: Texas Instruments Incorporated.
- Texas Instruments. (2009a). *SLAS554H: MSP430 SoC with RF Core Datasheet*. Dallas: Texas Instruments Incorporated.
- Texas Instruments. (2009b). *SLAU7259E: CC430 Family User's Guide*. Dallas: Texas Instruments Incorporated.
- Texas Instruments. (2009c). *SLAU728T: MSP430 Hardware Tools User's Guide*. Dallas: Texas Instruments Incorporated.

- Texas Instruments. (2009d). *SWRU222: CC USB Software Examples User's Guide*. Dallas: Texas Instruments Incorporated.
- Texas Instruments. (2014a). *CC1110Fx/CC1111Fx Family*. User Guide. Dallas: Texas Instruments Incorporated.
- Texas Instruments. (2014b). *SWRS033H: Low-Power SoC (System-on-Chip) with MCU, Memory, Sub-1 GHz Transceiver, and USB Controller*. Dallas: Texas Instruments Incorporated.
- Underwood, S. Frangline, J. & Chan, V. (2008). *Three-Phase Electronic Watt-Hour Meter Design Using MSP430*. Dallas: Texas Instruments Incorporated.
- Wines, M. (2007). *Application Note N049: Software for CC1100/CC2500 and MSP430*. Dallas: Texas Instruments Incorporated.

ANEXO A 1.2: Diagrama PCB EM430F6137RF900



*Tomado de: SLAU278T (Texas Instruments, 2009).

Figura 46: Diagrama PCB EM430F6137RF900

ANEXO B 1.2: Diagrama de bloques modo autoscan

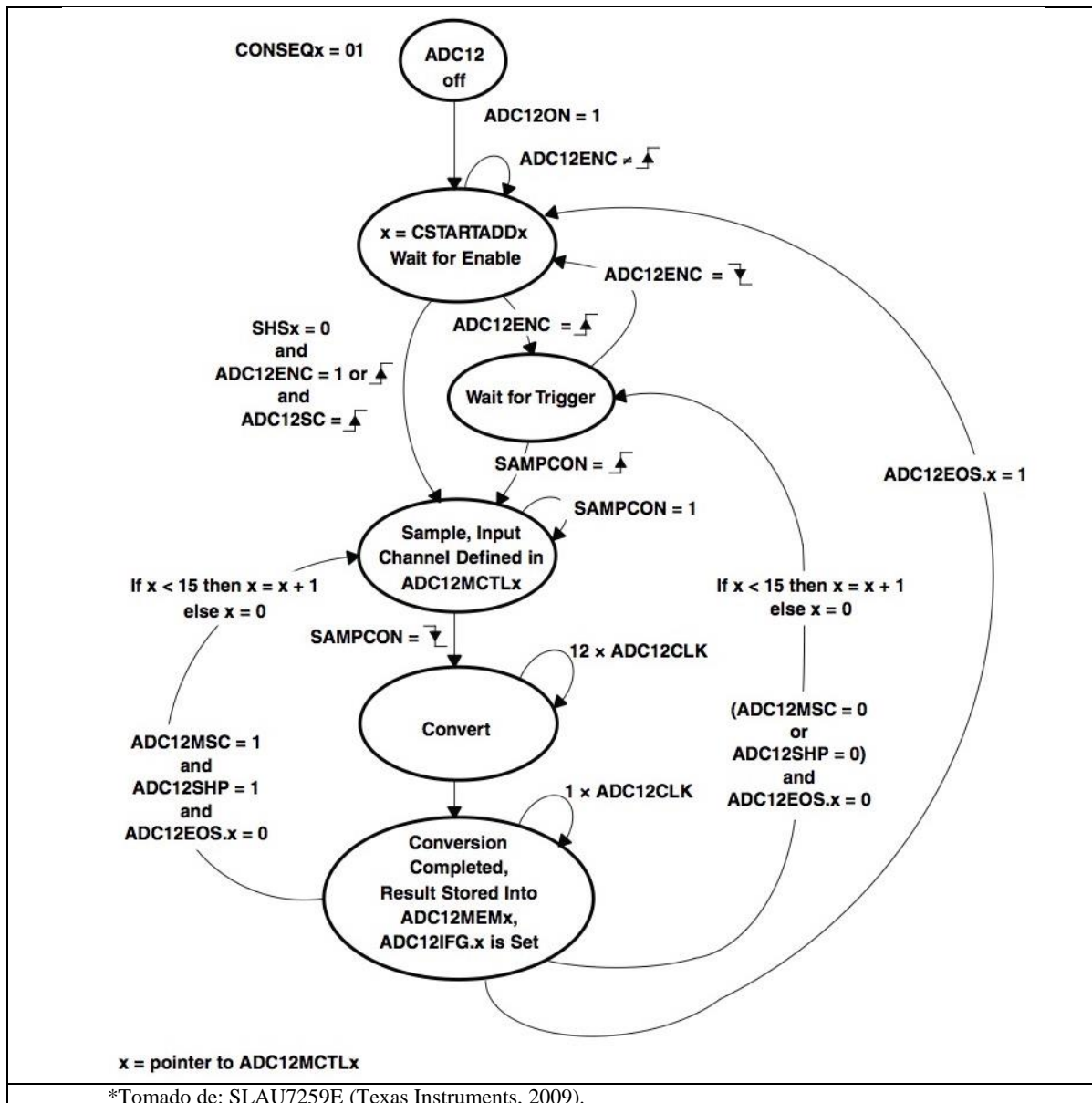
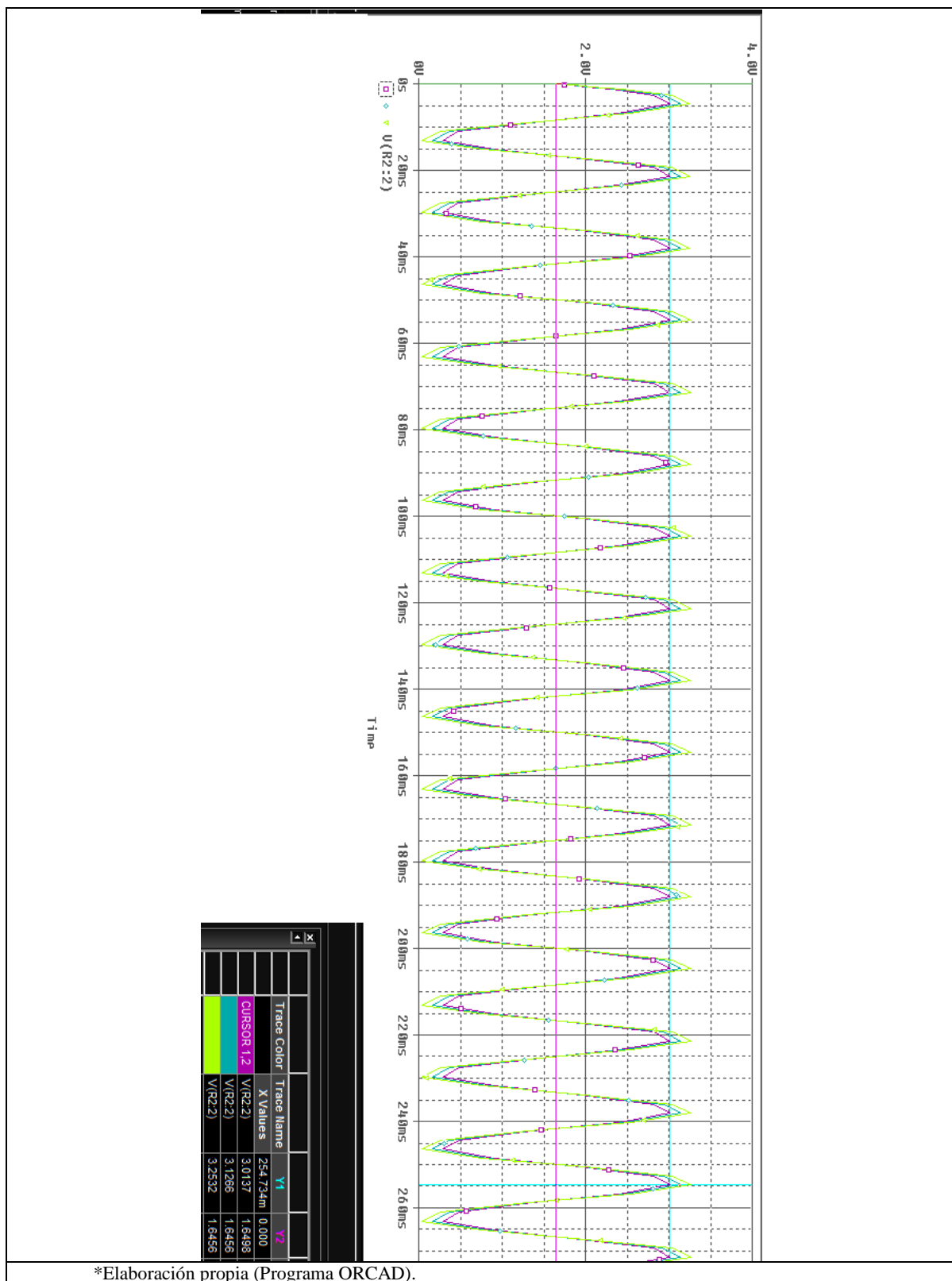


Figura 48: Modo Autoscan

ANEXO C 1.1: Simulación del sensor de voltaje



*Elaboración propia (Programa ORCAD).

Figura 49: Simulación Sensor Voltaje

ANEXO C 1.2: Diagrama de bloques ACS712-20A-T

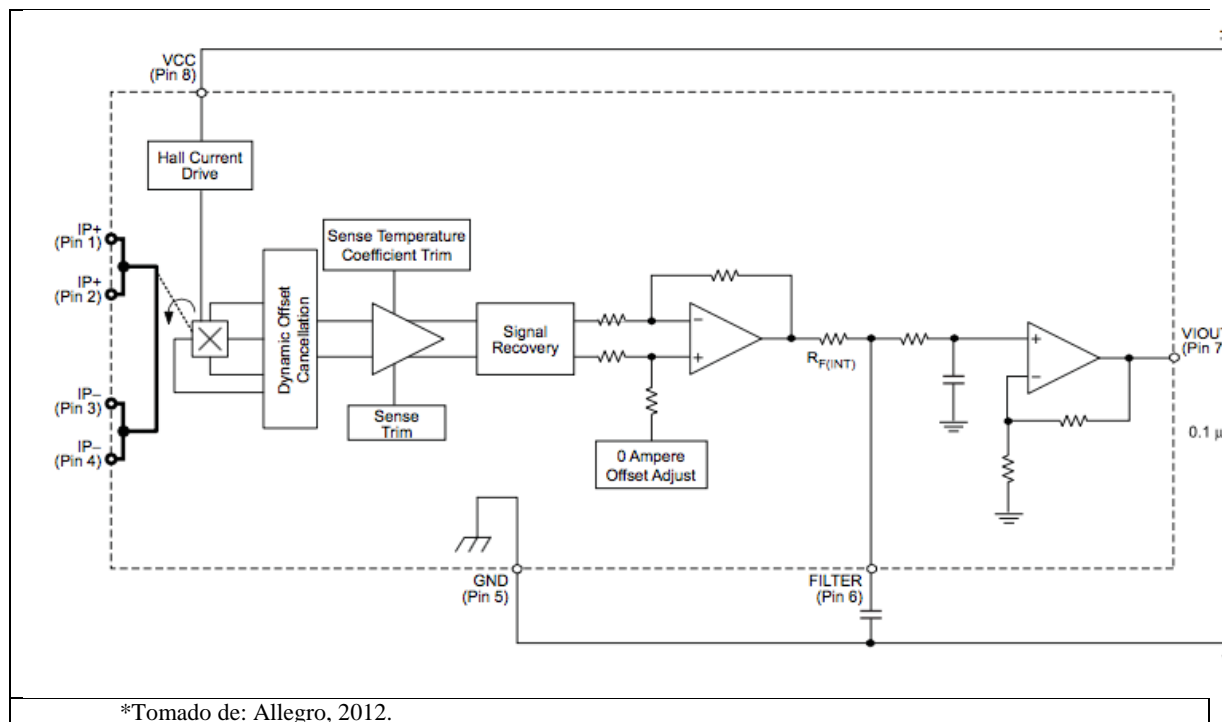


Figura 50: ACS712-20A-T

ANEXO C 1.3: Script en MATLAB para el dimensionamiento del sensor de corriente

```

%% Current Sensor Design
clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% vout = (1/R1+R21) * [Vn_inv*(R21+R4+R1)-(R4*Vin)]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Valores iniciales dados
R4 = 47e3;
R1 = 1e3;

% Valores variables convergentes
% Vn_inv = Voltaje entrada no inversora [0 - 5] [V]
% R21 = Resistencia de entrada inversora [0 - 100] [kohm]
% Vin = Voltaje entrada sensor corriente [0.6 - 4.6] [V]
% Vout = Voltaje de salida sensor [0 - 3.3] [V]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Ecuacion de ajuste %%%%%%%%%
% vout = -0.825Vin + 3.795
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Ecuacion de ajuste %%%%%%%%%

% Convergencia por pendiente
Vn_inv = (759/365); % condicion inicial
R21 = 50e3; % Valor inicial
ep = 1e2;
error = 1;
while abs(error) > 1e-3

    temp = (R4/(R1+R21));
    error = temp - 0.825;

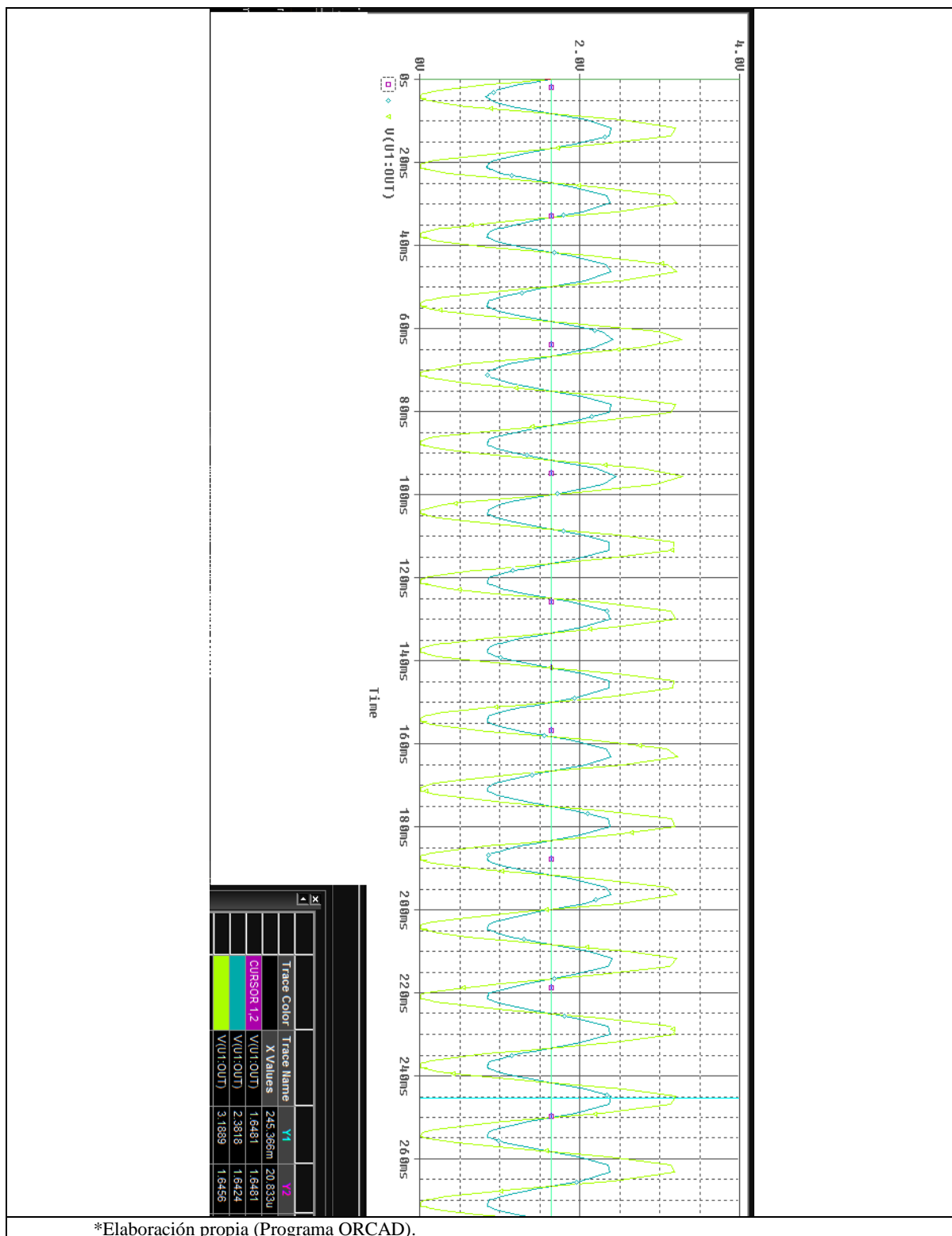
    if error > 0
        R21 = R21 + ep;
    end

    if error < 0
        R21 = R21 - ep;
    end
end

% Convergencia por factor b
ep = 1e-3;
error = 1;
while error > 1e-4
    temp = ((Vn_inv*(R21+R4+R1))/(R1+R21));
    error = temp - 3.795;
    if error > 0
        Vn_inv = Vn_inv - ep;
    end
    if error < 0
        Vn_inv = Vn_inv + ep;
    end
end

```

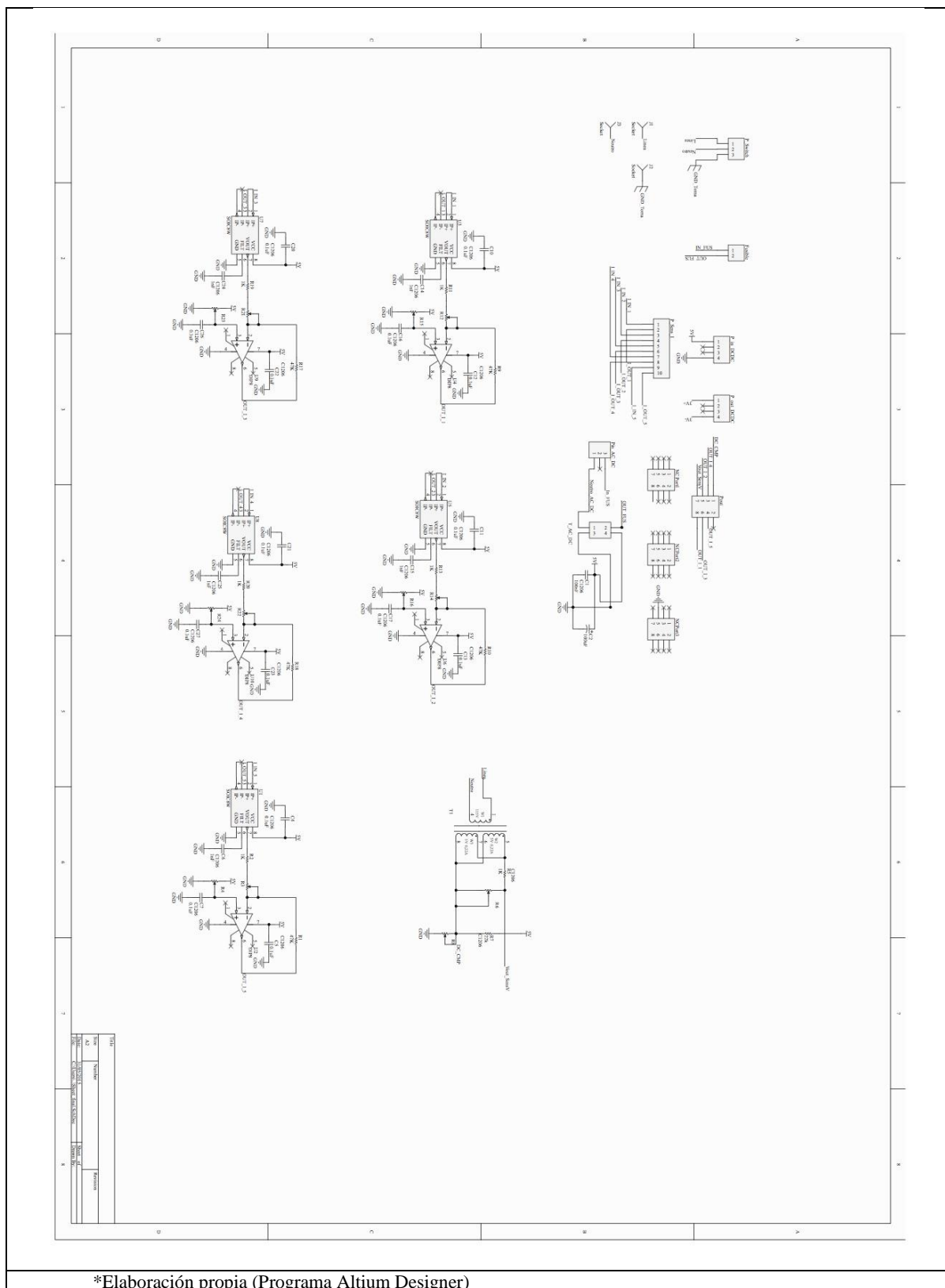
ANEXO C 1.4: Simulación del sensor de corriente



*Elaboración propia (Programa ORCAD).

Figura 51: Simulación Sensor Corriente

ANEXO D 1.1: Esquemático Final PCB



*Elaboración propia (Programa Altium Designer)

Figura 52: Esquemático PCB

ANEXO E 1.1: Algoritmo implementación filtro FIR en Matlab

```

% Generacion del Filtro FIR (Single-Tap zero-FIR Filter)
clear all
close all
clc
%%%%%%%%%%%% Ecuacion %%%%%%%%%%%%%
% y[n] = x[n] + beta*x[n-1]
%%%%%%%%%%%% Ecuacion %%%%%%%%%%%%%

fprintf('-----\n');
fprintf('Single-Tap zero-FIR Filter Calculation\n');
fprintf('Please Insert Input Parameters: \n');
fprintf('-----\n\n');

% w -> Angular Frecuency after sampling
% m_freq -> mains frecuency
% samp_freq -> sample frecuency
m_freq = input('Main Frecuency [Hz]: ');
samp_freq = input('Sample Frecuency [sample/s]: ');
% samp_freq = 3276.8;
w = 2*pi*(m_freq/samp_freq);

% FIR Parameters Table
limit_p = input('Max/Min Phase Shift [deg]: ');
tol_err = input('Max Error tolerance [%]: ');
min_pf = input('Min Power Factor: ');
pf_angle = acos(min_pf); %[rad]
zero_p = (limit_p/64)*(pi/180); %Meter table 128 tap (64 + / 64 - )

fprintf('-----\n');
fprintf('\nResults: \n');
fprintf('Phase Shift Optimal Step: %2.4f [deg]\n', zero_p*(180/pi));
fprintf('Max Error: %2.4f%% \n', tol_err);
fprintf('-----\n\n');

% Phase Shift vector
limit_p = limit_p*(pi/180); %[rad]
step = 0; % Inicial step value of vector
index = 1; % Filling index of vector

while step <= limit_p
    ZERO_POS(index) = step;
    step = step + zero_p;
    index = index + 1;
end

step = -1*zero_p; % Reset step value for negative steps
index = 1; % Reset index value for negative vector

while step >= -limit_p
    ZERO_NE(index) = step;
    step = step - zero_p;
    index = index + 1;
end

```

```

ZERO_POS = ZERO_POS(end:-1:1); % Flip vector position
ZERO_P = [ZERO_POS ZERO_NE];

% Coeficientes fir_coeffs
offset_beta = samp_freq*10; % Samp_freq = SAMP_FREQ_10_SEC / 10
offset_gain = offset_beta/2; % Punto medio (+ and - half sample)

for i = 1:length(ZERO_P)
% Calculo de D
D = ZERO_P(i)/w;

% Calculo de Beta
beta_1_num = ((1-2*D)*cos(w)) + sqrt((((1-2*D)^2)*cos(w)^2+4*D*(1-D));
beta_2_num = ((1-2*D)*cos(w)) - sqrt((((1-2*D)^2)*cos(w)^2+4*D*(1-D));
beta_1_den = 2*(1-D);
beta_2_den = 2*(1-D);

beta_1 = -1*(beta_1_num/beta_1_den);
beta_2 = -1*(beta_2_num/beta_2_den);

% Calculo de A (gain)
inv_A1 = ((cos(w)+beta_1)^2+sin(w)^2)^(-1/2);
inv_A2 = ((cos(w)+beta_2)^2+sin(w)^2)^(-1/2);

A_1 = 1/inv_A1;
A_2 = 1/inv_A2;

fir_beta(i) = int16(beta_2 * offset_beta);
fir_gain(i) = int16(offset_gain/A_2);

    if i == 1
    % Escritura del .txt
    fileID = fopen('emeter-fir-table-matlab.txt','w');
    fprintf(fileID,'\nconst int16_t fir_coeffs[][2] =\n');
    fprintf(fileID,'\n');
    fprintf(fileID,'\t{ %5hi, %5hi }, // %5.4f
deg\n',fir_beta(i),fir_gain(i),(180/pi)*ZERO_P(i));
    fclose(fileID);
    else
    fileID = fopen('emeter-fir-table-matlab.txt','a');
    fprintf(fileID,'\t{ %5hi, %5hi }, // %5.4f
deg\n',fir_beta(i),fir_gain(i),(180/pi)*ZERO_P(i));
    fclose(fileID);
    end
    if i == length(ZERO_P)
        fileID = fopen('emeter-fir-table-matlab.txt','a');
        fprintf(fileID,'};\n');
        fclose(fileID);
    end
end

fprintf('-----\n');
fprintf('FIR Table .txt created \n');
fprintf('-----\n');

```

ANEXO F 1.1: Mediciones por canal del sistema DAQ

Tabla 16: Mediciones Canal_2 (promedios)

V_real_2 [V]	V_rms_2 [V]	I_real_2 [A]	I_rms_2 [A]	P_real_2 [W]	P_Active_2 [W]	Factor_P_2	Frecuency_2 [Hz]
114.90	112.60	0.39	0.60	44.81	56.96	0.83	60.01
113.10	112.29	0.74	1.05	83.69	103.80	0.88	59.98
113.10	111.78	1.13	1.56	127.80	156.40	0.89	60.00
113.40	112.72	1.38	1.89	156.49	190.85	0.89	59.98
113.10	111.29	1.65	2.25	186.62	226.18	0.89	59.98
114.90	112.56	2.07	2.80	237.84	285.35	0.89	59.99
113.70	112.03	2.47	3.33	280.84	336.96	0.89	60.00
113.90	112.20	2.87	3.88	326.89	393.68	0.89	60.01
113.70	111.73	3.16	4.27	359.29	431.93	0.89	60.00
114.60	112.42	3.42	4.59	391.93	467.58	0.89	60.00
114.80	112.59	3.85	5.14	441.98	521.46	0.89	60.00
113.80	111.83	4.22	5.69	480.24	579.22	0.89	59.96
112.90	110.55	4.79	6.46	540.79	651.79	0.90	59.98
113.40	111.35	5.15	6.89	584.01	695.62	0.90	60.00
115.80	113.93	5.65	7.59	654.27	780.55	0.89	59.97

Tabla 17: Mediciones Canal_3 (promedios)

V_real_3 [V]	V_rms_3 [V]	I_real_3 [A]	I_rms_3 [A]	P_real_3 [W]	P_Active_3 [W]	Factor_P_3	Frecuency_3 [Hz]
114.90	113.51	0.39	0.61	44.81	56.59	0.81	60.01
113.10	111.48	0.74	1.05	83.69	103.26	0.87	59.98
113.10	111.25	1.13	1.56	127.80	154.77	0.88	60.00
113.40	111.54	1.38	1.89	156.49	189.73	0.89	59.98
113.10	112.25	1.65	2.25	186.62	225.74	0.89	59.98
114.90	112.92	2.07	2.79	237.84	283.66	0.89	60.00
113.70	111.49	2.47	3.31	280.84	333.73	0.89	60.00
113.90	112.61	2.87	3.86	326.89	391.31	0.89	60.02
113.70	112.22	3.16	4.25	359.29	429.47	0.89	60.01
114.60	111.97	3.42	4.57	391.93	464.69	0.89	60.00
114.80	112.35	3.85	5.12	441.98	519.06	0.89	60.00
113.80	112.57	4.22	5.68	480.24	577.03	0.89	59.96
112.90	110.90	4.79	6.42	540.79	647.40	0.90	59.99
113.40	111.53	5.15	6.87	584.01	693.74	0.90	60.00
115.80	112.95	5.65	7.56	654.27	776.06	0.89	59.97

Tabla 18: Mediciones Canal_4 (promedios)

V_real_4 [V]	V_rms_4 [V]	I_real_4 [A]	I_rms_4 [A]	P_real_4 [W]	P_Active_4 [W]	Factor_P_4	Frecuency_4 [Hz]
114.90	113.85	0.39	0.61	44.81	55.38	0.79	60.01
113.10	111.65	0.74	1.04	83.69	101.48	0.86	59.97
113.10	111.77	1.13	1.55	127.80	153.12	0.88	60.00
113.40	111.91	1.38	1.87	156.49	187.41	0.88	59.97
113.10	112.30	1.65	2.23	186.62	223.51	0.89	59.99
114.90	113.38	2.07	2.77	237.84	281.42	0.89	60.00
113.70	111.67	2.47	3.29	280.84	331.59	0.89	60.00
113.90	111.68	2.87	3.83	326.89	387.61	0.89	60.01
113.70	111.56	3.16	4.22	359.29	425.46	0.89	60.01
114.60	111.90	3.42	4.55	391.93	462.14	0.89	60.01
114.80	112.31	3.85	5.10	441.98	517.25	0.89	60.00
113.80	112.78	4.22	5.62	480.24	570.77	0.89	59.96
112.90	111.44	4.79	6.38	540.79	642.92	0.90	59.98
113.40	110.65	5.15	6.81	584.01	686.59	0.90	60.01
115.80	113.65	5.65	7.52	654.27	772.15	0.89	59.97

Tabla 19: Mediciones Canal_5 (promedios)

V_real_5 [V]	V_rms_5 [V]	I_real_5 [A]	I_rms_5 [A]	P_real_5 [W]	P_Active_5 [W]	Factor_P_5	Frecuency_5 [Hz]
115.10	114.41	0.00	0.32	0.00	5.64	0.15	60.01
114.90	113.08	0.39	0.60	44.81	49.67	0.73	60.01
113.10	112.43	0.74	1.02	83.69	96.03	0.83	59.99
113.10	110.43	1.13	1.52	127.80	147.55	0.86	59.99
113.40	111.39	1.38	1.84	156.49	181.40	0.87	59.97
113.10	111.37	1.65	2.20	186.62	217.28	0.88	59.98
114.90	113.16	2.07	2.73	237.84	274.99	0.88	60.00
113.70	110.46	2.47	3.25	280.84	324.80	0.88	60.00
113.90	111.42	2.87	3.79	326.89	381.02	0.88	60.02
113.70	111.62	3.16	4.18	359.29	419.43	0.89	60.00
114.60	111.94	3.42	4.50	391.93	454.48	0.89	60.00
114.80	111.88	3.85	5.04	441.98	507.79	0.89	59.99
113.80	112.69	4.22	5.59	480.24	565.83	0.89	59.96
112.90	111.69	4.79	6.34	540.79	636.71	0.89	59.99
113.40	110.02	5.15	6.78	584.01	681.92	0.89	59.99
115.80	113.06	5.65	7.46	654.27	764.81	0.89	59.97

ANEXO F 1.2: Pruebas t para las mediciones del sistema DAQ

Tabla 20: Pruebas t-pareadas

Canal	Voltaje	Corriente	Potencia																																																												
1	Paired T-Test and CI: V_real_1 [V], V_rms_1 [V] Paired T for V_real_1 [V] - V_rms_1 [V] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>V_real_1 [V]</td> <td>15</td> <td>113.940</td> <td>0.862</td> <td>0.222</td> </tr> <tr> <td>V_rms_1 [V]</td> <td>15</td> <td>112.931</td> <td>0.705</td> <td>0.182</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>1.0092</td> <td>0.3246</td> <td>0.0838</td> </tr> </tbody> </table> 95% CI for mean difference: (0.8295, 1.1890) T-Test of mean difference = 0 (vs ≠ 0): T-Value = 12.04 P-Value = 0.000		N	Mean	StDev	SE Mean	V_real_1 [V]	15	113.940	0.862	0.222	V_rms_1 [V]	15	112.931	0.705	0.182	Difference	15	1.0092	0.3246	0.0838	Paired T-Test and CI: I_real_1 [A], I_rms_1 [A] Paired T for I_real_1 [A] - I_rms_1 [A] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>I_real_1 [A]</td> <td>15</td> <td>2.863</td> <td>1.648</td> <td>0.425</td> </tr> <tr> <td>I_rms_1 [A]</td> <td>15</td> <td>4.837</td> <td>2.742</td> <td>0.708</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>-1.974</td> <td>1.095</td> <td>0.283</td> </tr> </tbody> </table> 95% CI for mean difference: (-2.581, -1.368) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -6.99 P-Value = 0.000		N	Mean	StDev	SE Mean	I_real_1 [A]	15	2.863	1.648	0.425	I_rms_1 [A]	15	4.837	2.742	0.708	Difference	15	-1.974	1.095	0.283	Paired T-Test and CI: P_real_1 [W], P_Active_1 [W] Paired T for P_real_1 [W] - P_Active_1 [W] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>P_real_1 [W]</td> <td>15</td> <td>326.5</td> <td>188.6</td> <td>48.7</td> </tr> <tr> <td>P_Active_1 [W]</td> <td>15</td> <td>490.7</td> <td>282.2</td> <td>72.9</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>-164.2</td> <td>93.6</td> <td>24.2</td> </tr> </tbody> </table> 95% CI for mean difference: (-216.1, -112.4) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -6.80 P-Value = 0.000		N	Mean	StDev	SE Mean	P_real_1 [W]	15	326.5	188.6	48.7	P_Active_1 [W]	15	490.7	282.2	72.9	Difference	15	-164.2	93.6	24.2
		N	Mean	StDev	SE Mean																																																										
V_real_1 [V]	15	113.940	0.862	0.222																																																											
V_rms_1 [V]	15	112.931	0.705	0.182																																																											
Difference	15	1.0092	0.3246	0.0838																																																											
	N	Mean	StDev	SE Mean																																																											
I_real_1 [A]	15	2.863	1.648	0.425																																																											
I_rms_1 [A]	15	4.837	2.742	0.708																																																											
Difference	15	-1.974	1.095	0.283																																																											
	N	Mean	StDev	SE Mean																																																											
P_real_1 [W]	15	326.5	188.6	48.7																																																											
P_Active_1 [W]	15	490.7	282.2	72.9																																																											
Difference	15	-164.2	93.6	24.2																																																											
2	Paired T-Test and CI: V_real_2 [V], V_rms_2 [V] Paired T for V_real_2 [V] - V_rms_2 [V] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>V_real_2 [V]</td> <td>15</td> <td>113.940</td> <td>0.862</td> <td>0.222</td> </tr> <tr> <td>V_rms_2 [V]</td> <td>15</td> <td>112.124</td> <td>0.779</td> <td>0.201</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>1.816</td> <td>0.519</td> <td>0.134</td> </tr> </tbody> </table> 95% CI for mean difference: (1.529, 2.104) T-Test of mean difference = 0 (vs ≠ 0): T-Value = 13.56 P-Value = 0.000		N	Mean	StDev	SE Mean	V_real_2 [V]	15	113.940	0.862	0.222	V_rms_2 [V]	15	112.124	0.779	0.201	Difference	15	1.816	0.519	0.134	Paired T-Test and CI: I_real_2 [A], I_rms_2 [A] Paired T for I_real_2 [A] - I_rms_2 [A] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>I_real_2 [A]</td> <td>15</td> <td>2.863</td> <td>1.648</td> <td>0.425</td> </tr> <tr> <td>I_rms_2 [A]</td> <td>15</td> <td>3.866</td> <td>2.192</td> <td>0.566</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>-1.003</td> <td>0.545</td> <td>0.141</td> </tr> </tbody> </table> 95% CI for mean difference: (-1.305, -0.702) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -7.14 P-Value = 0.000		N	Mean	StDev	SE Mean	I_real_2 [A]	15	2.863	1.648	0.425	I_rms_2 [A]	15	3.866	2.192	0.566	Difference	15	-1.003	0.545	0.141	Paired T-Test and CI: P_real_2 [W], P_Active_2 [W] Paired T for P_real_2 [W] - P_Active_2 [W] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>P_real_2 [W]</td> <td>15</td> <td>326.5</td> <td>188.6</td> <td>48.7</td> </tr> <tr> <td>P_Active_2 [W]</td> <td>15</td> <td>391.9</td> <td>224.1</td> <td>57.9</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>-65.39</td> <td>35.54</td> <td>9.18</td> </tr> </tbody> </table> 95% CI for mean difference: (-85.07, -45.71) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -7.13 P-Value = 0.000		N	Mean	StDev	SE Mean	P_real_2 [W]	15	326.5	188.6	48.7	P_Active_2 [W]	15	391.9	224.1	57.9	Difference	15	-65.39	35.54	9.18
		N	Mean	StDev	SE Mean																																																										
V_real_2 [V]	15	113.940	0.862	0.222																																																											
V_rms_2 [V]	15	112.124	0.779	0.201																																																											
Difference	15	1.816	0.519	0.134																																																											
	N	Mean	StDev	SE Mean																																																											
I_real_2 [A]	15	2.863	1.648	0.425																																																											
I_rms_2 [A]	15	3.866	2.192	0.566																																																											
Difference	15	-1.003	0.545	0.141																																																											
	N	Mean	StDev	SE Mean																																																											
P_real_2 [W]	15	326.5	188.6	48.7																																																											
P_Active_2 [W]	15	391.9	224.1	57.9																																																											
Difference	15	-65.39	35.54	9.18																																																											
3	Paired T-Test and CI: V_real_3 [V], V_rms_3 [V] Paired T for V_real_3 [V] - V_rms_3 [V] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>V_real_3 [V]</td> <td>15</td> <td>113.940</td> <td>0.862</td> <td>0.222</td> </tr> <tr> <td>V_rms_3 [V]</td> <td>15</td> <td>112.101</td> <td>0.735</td> <td>0.190</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>1.839</td> <td>0.550</td> <td>0.142</td> </tr> </tbody> </table> 95% CI for mean difference: (1.535, 2.143) T-Test of mean difference = 0 (vs ≠ 0): T-Value = 12.96 P-Value = 0.000		N	Mean	StDev	SE Mean	V_real_3 [V]	15	113.940	0.862	0.222	V_rms_3 [V]	15	112.101	0.735	0.190	Difference	15	1.839	0.550	0.142	Paired T-Test and CI: I_real_3 [A], I_rms_3 [A] Paired T for I_real_3 [A] - I_rms_3 [A] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>I_real_3 [A]</td> <td>15</td> <td>2.863</td> <td>1.648</td> <td>0.425</td> </tr> <tr> <td>I_rms_3 [A]</td> <td>15</td> <td>3.851</td> <td>2.180</td> <td>0.563</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>-0.988</td> <td>0.533</td> <td>0.138</td> </tr> </tbody> </table> 95% CI for mean difference: (-1.283, -0.693) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -7.19 P-Value = 0.000		N	Mean	StDev	SE Mean	I_real_3 [A]	15	2.863	1.648	0.425	I_rms_3 [A]	15	3.851	2.180	0.563	Difference	15	-0.988	0.533	0.138	Paired T-Test and CI: P_real_3 [W], P_Active_3 [W] Paired T for P_real_3 [W] - P_Active_3 [W] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>P_real_3 [W]</td> <td>15</td> <td>326.5</td> <td>188.6</td> <td>48.7</td> </tr> <tr> <td>P_Active_3 [W]</td> <td>15</td> <td>389.7</td> <td>223.1</td> <td>57.6</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>-63.25</td> <td>34.53</td> <td>8.92</td> </tr> </tbody> </table> 95% CI for mean difference: (-82.37, -44.13) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -7.09 P-Value = 0.000		N	Mean	StDev	SE Mean	P_real_3 [W]	15	326.5	188.6	48.7	P_Active_3 [W]	15	389.7	223.1	57.6	Difference	15	-63.25	34.53	8.92
		N	Mean	StDev	SE Mean																																																										
V_real_3 [V]	15	113.940	0.862	0.222																																																											
V_rms_3 [V]	15	112.101	0.735	0.190																																																											
Difference	15	1.839	0.550	0.142																																																											
	N	Mean	StDev	SE Mean																																																											
I_real_3 [A]	15	2.863	1.648	0.425																																																											
I_rms_3 [A]	15	3.851	2.180	0.563																																																											
Difference	15	-0.988	0.533	0.138																																																											
	N	Mean	StDev	SE Mean																																																											
P_real_3 [W]	15	326.5	188.6	48.7																																																											
P_Active_3 [W]	15	389.7	223.1	57.6																																																											
Difference	15	-63.25	34.53	8.92																																																											
4	Paired T-Test and CI: V_real_4 [V], V_rms_4 [V] Paired T for V_real_4 [V] - V_rms_4 [V] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>V_real_4 [V]</td> <td>15</td> <td>113.940</td> <td>0.862</td> <td>0.222</td> </tr> <tr> <td>V_rms_4 [V]</td> <td>15</td> <td>112.167</td> <td>0.894</td> <td>0.231</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>1.773</td> <td>0.624</td> <td>0.161</td> </tr> </tbody> </table> 95% CI for mean difference: (1.428, 2.118) T-Test of mean difference = 0 (vs ≠ 0): T-Value = 11.01 P-Value = 0.000		N	Mean	StDev	SE Mean	V_real_4 [V]	15	113.940	0.862	0.222	V_rms_4 [V]	15	112.167	0.894	0.231	Difference	15	1.773	0.624	0.161	Paired T-Test and CI: I_real_4 [A], I_rms_4 [A] Paired T for I_real_4 [A] - I_rms_4 [A] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>I_real_4 [A]</td> <td>15</td> <td>2.863</td> <td>1.648</td> <td>0.425</td> </tr> <tr> <td>I_rms_4 [A]</td> <td>15</td> <td>3.827</td> <td>2.165</td> <td>0.559</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>-0.964</td> <td>0.517</td> <td>0.134</td> </tr> </tbody> </table> 95% CI for mean difference: (-1.251, -0.678) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -7.22 P-Value = 0.000		N	Mean	StDev	SE Mean	I_real_4 [A]	15	2.863	1.648	0.425	I_rms_4 [A]	15	3.827	2.165	0.559	Difference	15	-0.964	0.517	0.134	Paired T-Test and CI: P_real_4 [W], P_Active_4 [W] Paired T for P_real_4 [W] - P_Active_4 [W] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>P_real_4 [W]</td> <td>15</td> <td>326.5</td> <td>188.6</td> <td>48.7</td> </tr> <tr> <td>P_Active_4 [W]</td> <td>15</td> <td>386.6</td> <td>221.8</td> <td>57.3</td> </tr> <tr> <td>Difference</td> <td>15</td> <td>-60.09</td> <td>33.18</td> <td>8.57</td> </tr> </tbody> </table> 95% CI for mean difference: (-78.46, -41.71) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -7.01 P-Value = 0.000		N	Mean	StDev	SE Mean	P_real_4 [W]	15	326.5	188.6	48.7	P_Active_4 [W]	15	386.6	221.8	57.3	Difference	15	-60.09	33.18	8.57
		N	Mean	StDev	SE Mean																																																										
V_real_4 [V]	15	113.940	0.862	0.222																																																											
V_rms_4 [V]	15	112.167	0.894	0.231																																																											
Difference	15	1.773	0.624	0.161																																																											
	N	Mean	StDev	SE Mean																																																											
I_real_4 [A]	15	2.863	1.648	0.425																																																											
I_rms_4 [A]	15	3.827	2.165	0.559																																																											
Difference	15	-0.964	0.517	0.134																																																											
	N	Mean	StDev	SE Mean																																																											
P_real_4 [W]	15	326.5	188.6	48.7																																																											
P_Active_4 [W]	15	386.6	221.8	57.3																																																											
Difference	15	-60.09	33.18	8.57																																																											
5	Paired T-Test and CI: V_real_5 [V], V_rms_5 [V] Paired T for V_real_5 [V] - V_rms_5 [V] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>V_real_5 [V]</td> <td>16</td> <td>114.013</td> <td>0.882</td> <td>0.220</td> </tr> <tr> <td>V_rms_5 [V]</td> <td>16</td> <td>111.941</td> <td>1.159</td> <td>0.290</td> </tr> <tr> <td>Difference</td> <td>16</td> <td>2.071</td> <td>0.855</td> <td>0.214</td> </tr> </tbody> </table> 95% CI for mean difference: (1.616, 2.527) T-Test of mean difference = 0 (vs ≠ 0): T-Value = 9.69 P-Value = 0.000		N	Mean	StDev	SE Mean	V_real_5 [V]	16	114.013	0.882	0.220	V_rms_5 [V]	16	111.941	1.159	0.290	Difference	16	2.071	0.855	0.214	Paired T-Test and CI: I_real_5 [A], I_rms_5 [A] Paired T for I_real_5 [A] - I_rms_5 [A] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>I_real_5 [A]</td> <td>16</td> <td>2.684</td> <td>1.745</td> <td>0.436</td> </tr> <tr> <td>I_rms_5 [A]</td> <td>16</td> <td>3.571</td> <td>2.257</td> <td>0.564</td> </tr> <tr> <td>Difference</td> <td>16</td> <td>-0.887</td> <td>0.515</td> <td>0.129</td> </tr> </tbody> </table> 95% CI for mean difference: (-1.162, -0.613) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -6.89 P-Value = 0.000		N	Mean	StDev	SE Mean	I_real_5 [A]	16	2.684	1.745	0.436	I_rms_5 [A]	16	3.571	2.257	0.564	Difference	16	-0.887	0.515	0.129	Paired T-Test and CI: P_real_5 [W], P_Active_5 [W] Paired T for P_real_5 [W] - P_Active_5 [W] <table border="1"> <thead> <tr> <th></th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> </tr> </thead> <tbody> <tr> <td>P_real_5 [W]</td> <td>16</td> <td>306.1</td> <td>199.7</td> <td>49.9</td> </tr> <tr> <td>P_Active_5 [W]</td> <td>16</td> <td>356.8</td> <td>233.6</td> <td>58.4</td> </tr> <tr> <td>Difference</td> <td>16</td> <td>-50.74</td> <td>34.06</td> <td>8.51</td> </tr> </tbody> </table> 95% CI for mean difference: (-68.89, -32.59) T-Test of mean difference = 0 (vs ≠ 0): T-Value = -5.96 P-Value = 0.000		N	Mean	StDev	SE Mean	P_real_5 [W]	16	306.1	199.7	49.9	P_Active_5 [W]	16	356.8	233.6	58.4	Difference	16	-50.74	34.06	8.51
		N	Mean	StDev	SE Mean																																																										
V_real_5 [V]	16	114.013	0.882	0.220																																																											
V_rms_5 [V]	16	111.941	1.159	0.290																																																											
Difference	16	2.071	0.855	0.214																																																											
	N	Mean	StDev	SE Mean																																																											
I_real_5 [A]	16	2.684	1.745	0.436																																																											
I_rms_5 [A]	16	3.571	2.257	0.564																																																											
Difference	16	-0.887	0.515	0.129																																																											
	N	Mean	StDev	SE Mean																																																											
P_real_5 [W]	16	306.1	199.7	49.9																																																											
P_Active_5 [W]	16	356.8	233.6	58.4																																																											
Difference	16	-50.74	34.06	8.51																																																											

Tabla 21: Pruebas t-de una muestra

Canal	Factor de Potencia	Frecuencia																																
1	One-Sample T: Factor_P_1 Test of $\mu = 1$ vs $\neq 1$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Factor_P_1</td> <td>15</td> <td>0.8786</td> <td>0.0538</td> <td>0.0139</td> <td>(0.8488, 0.9084)</td> <td>-8.74</td> <td>0.000</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Factor_P_1	15	0.8786	0.0538	0.0139	(0.8488, 0.9084)	-8.74	0.000	One-Sample T: Frecuency_1 [Hz] Test of $\mu = 60$ vs $\neq 60$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Frecuency_1 [Hz]</td> <td>15</td> <td>59.9923</td> <td>0.0170</td> <td>0.0044</td> <td>(59.9829, 60.0017)</td> <td>-1.75</td> <td>0.101</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Frecuency_1 [Hz]	15	59.9923	0.0170	0.0044	(59.9829, 60.0017)	-1.75	0.101
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Factor_P_1	15	0.8786	0.0538	0.0139	(0.8488, 0.9084)	-8.74	0.000																											
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Frecuency_1 [Hz]	15	59.9923	0.0170	0.0044	(59.9829, 60.0017)	-1.75	0.101																											
2	One-Sample T: Factor_P_2 Test of $\mu = 1$ vs $\neq 1$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Factor_P_2</td> <td>15</td> <td>0.88744</td> <td>0.01570</td> <td>0.00405</td> <td>(0.87875, 0.89614)</td> <td>-27.76</td> <td>0.000</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Factor_P_2	15	0.88744	0.01570	0.00405	(0.87875, 0.89614)	-27.76	0.000	One-Sample T: Frecuency_2 [Hz] Test of $\mu = 60$ vs $\neq 60$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Frecuency_2 [Hz]</td> <td>15</td> <td>59.9917</td> <td>0.0152</td> <td>0.0039</td> <td>(59.9833, 60.0001)</td> <td>-2.12</td> <td>0.052</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Frecuency_2 [Hz]	15	59.9917	0.0152	0.0039	(59.9833, 60.0001)	-2.12	0.052
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Factor_P_2	15	0.88744	0.01570	0.00405	(0.87875, 0.89614)	-27.76	0.000																											
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Frecuency_2 [Hz]	15	59.9917	0.0152	0.0039	(59.9833, 60.0001)	-2.12	0.052																											
3	One-Sample T: Factor_P_3 Test of $\mu = 1$ vs $\neq 1$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Factor_P_3</td> <td>15</td> <td>0.88391</td> <td>0.02185</td> <td>0.00564</td> <td>(0.87181, 0.89601)</td> <td>-20.58</td> <td>0.000</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Factor_P_3	15	0.88391	0.02185	0.00564	(0.87181, 0.89601)	-20.58	0.000	One-Sample T: Frecuency_3 [Hz] Test of $\mu = 60$ vs $\neq 60$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Frecuency_3 [Hz]</td> <td>15</td> <td>59.9927</td> <td>0.0153</td> <td>0.0039</td> <td>(59.9842, 60.0012)</td> <td>-1.85</td> <td>0.085</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Frecuency_3 [Hz]	15	59.9927	0.0153	0.0039	(59.9842, 60.0012)	-1.85	0.085
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Factor_P_3	15	0.88391	0.02185	0.00564	(0.87181, 0.89601)	-20.58	0.000																											
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Frecuency_3 [Hz]	15	59.9927	0.0153	0.0039	(59.9842, 60.0012)	-1.85	0.085																											
4	One-Sample T: Factor_P_4 Test of $\mu = 1$ vs $\neq 1$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Factor_P_4</td> <td>15</td> <td>0.88044</td> <td>0.02649</td> <td>0.00684</td> <td>(0.86578, 0.89511)</td> <td>-17.48</td> <td>0.000</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Factor_P_4	15	0.88044	0.02649	0.00684	(0.86578, 0.89511)	-17.48	0.000	One-Sample T: Frecuency_4 [Hz] Test of $\mu = 60$ vs $\neq 60$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Frecuency_4 [Hz]</td> <td>15</td> <td>59.9933</td> <td>0.0179</td> <td>0.0046</td> <td>(59.9833, 60.0032)</td> <td>-1.46</td> <td>0.166</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Frecuency_4 [Hz]	15	59.9933	0.0179	0.0046	(59.9833, 60.0032)	-1.46	0.166
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Factor_P_4	15	0.88044	0.02649	0.00684	(0.86578, 0.89511)	-17.48	0.000																											
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Frecuency_4 [Hz]	15	59.9933	0.0179	0.0046	(59.9833, 60.0032)	-1.46	0.166																											
5	One-Sample T: Factor_P_5 Test of $\mu = 1$ vs $\neq 1$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Factor_P_5</td> <td>16</td> <td>0.8245</td> <td>0.1836</td> <td>0.0459</td> <td>(0.7267, 0.9224)</td> <td>-3.82</td> <td>0.002</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Factor_P_5	16	0.8245	0.1836	0.0459	(0.7267, 0.9224)	-3.82	0.002	One-Sample T: Frecuency_5 [Hz] Test of $\mu = 60$ vs $\neq 60$ <table border="1"> <thead> <tr> <th>Variable</th> <th>N</th> <th>Mean</th> <th>StDev</th> <th>SE Mean</th> <th>95% CI</th> <th>T</th> <th>P</th> </tr> </thead> <tbody> <tr> <td>Frecuency_5 [Hz]</td> <td>16</td> <td>59.9928</td> <td>0.0163</td> <td>0.0041</td> <td>(59.9841, 60.0015)</td> <td>-1.77</td> <td>0.098</td> </tr> </tbody> </table>	Variable	N	Mean	StDev	SE Mean	95% CI	T	P	Frecuency_5 [Hz]	16	59.9928	0.0163	0.0041	(59.9841, 60.0015)	-1.77	0.098
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Factor_P_5	16	0.8245	0.1836	0.0459	(0.7267, 0.9224)	-3.82	0.002																											
Variable	N	Mean	StDev	SE Mean	95% CI	T	P																											
Frecuency_5 [Hz]	16	59.9928	0.0163	0.0041	(59.9841, 60.0015)	-1.77	0.098																											

ANEXO F 1.3: Gráficas de dispersión por canal del sistema DAQ

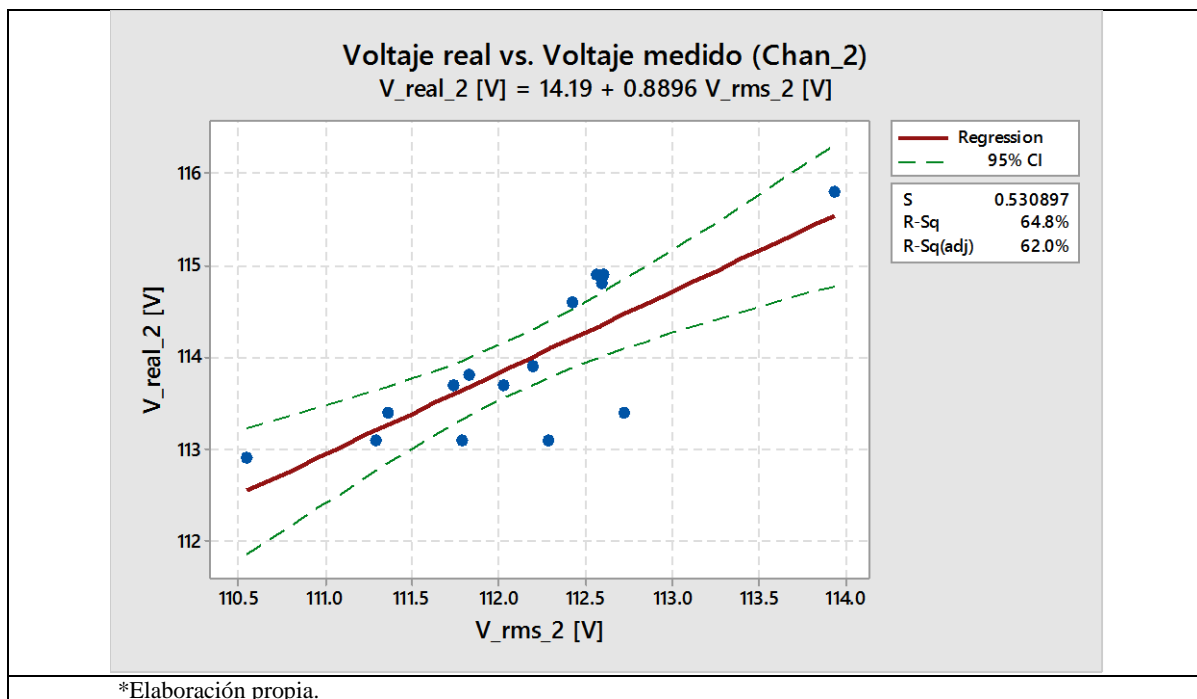


Figura 53: Gráfica de dispersión voltaje Chan_2

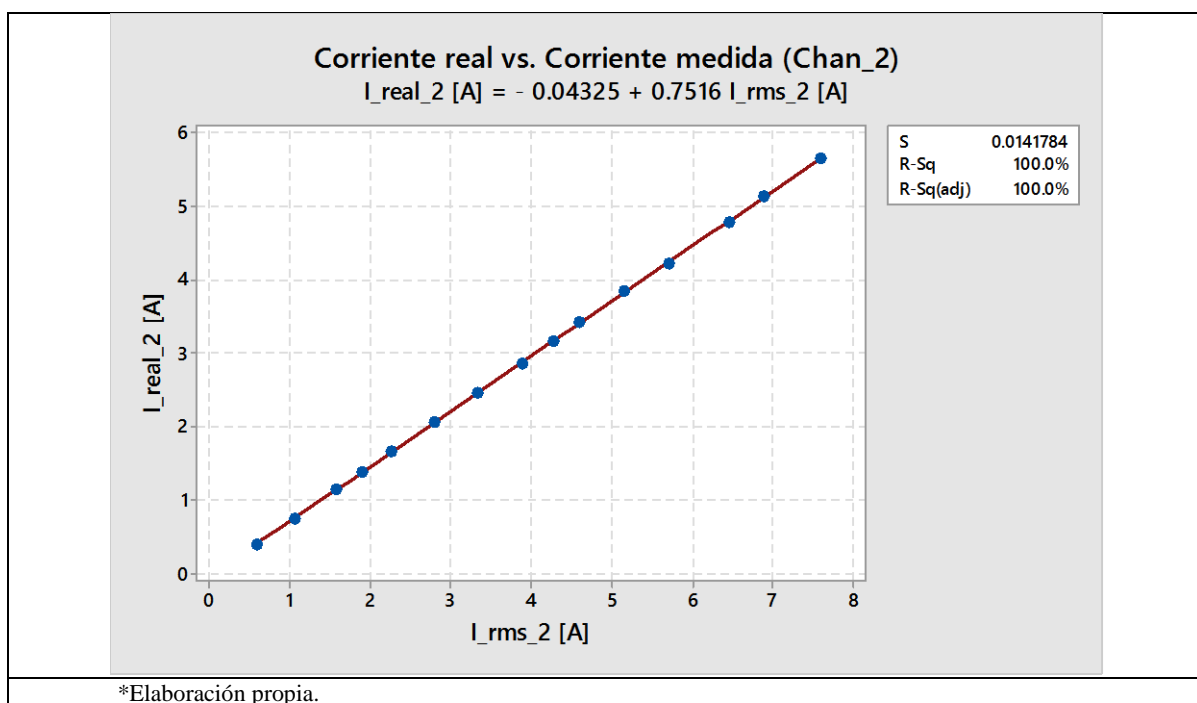


Figura 54: Gráfica de dispersión corriente Chan_2

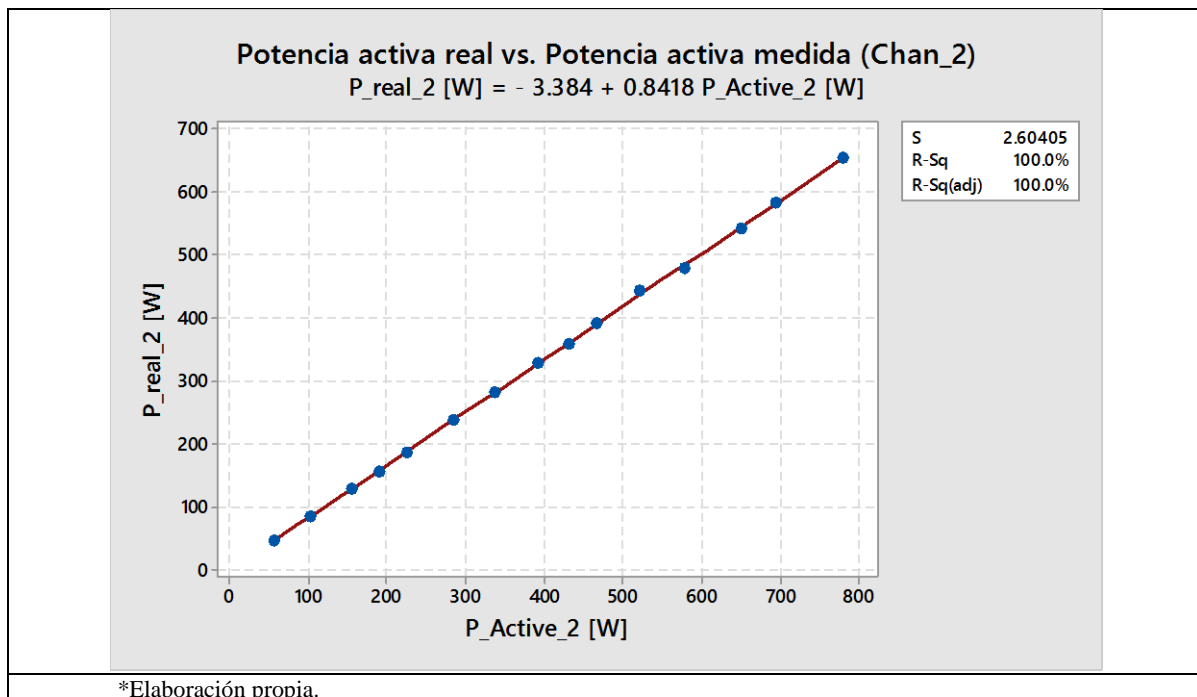


Figura 55: Gráfica de dispersión potencia activa Chan_2

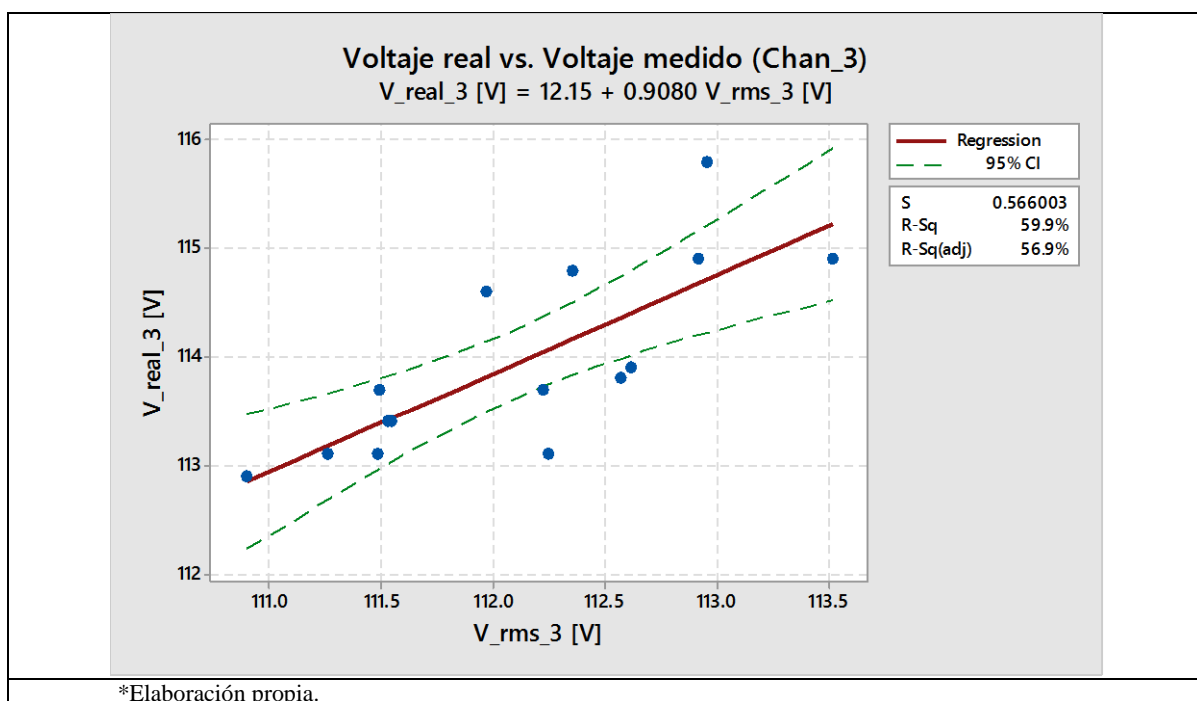


Figura 56: Gráfica de dispersión voltaje Chan_3

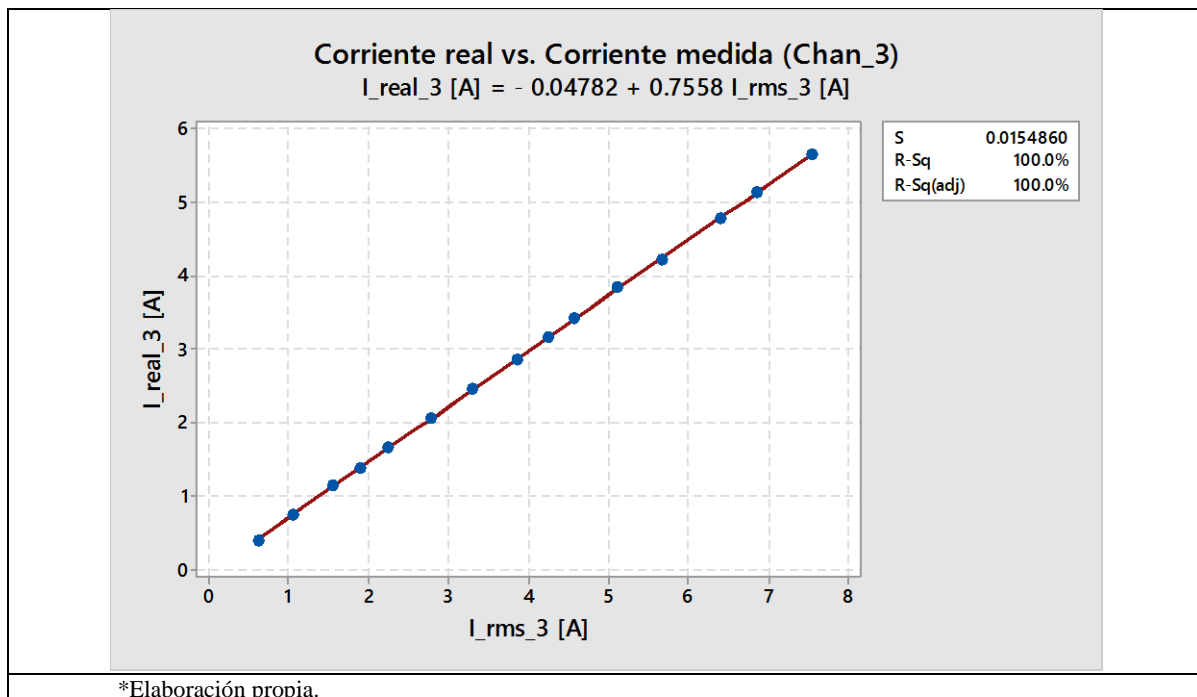


Figura 57: Gráfica de dispersión corriente Chan_3

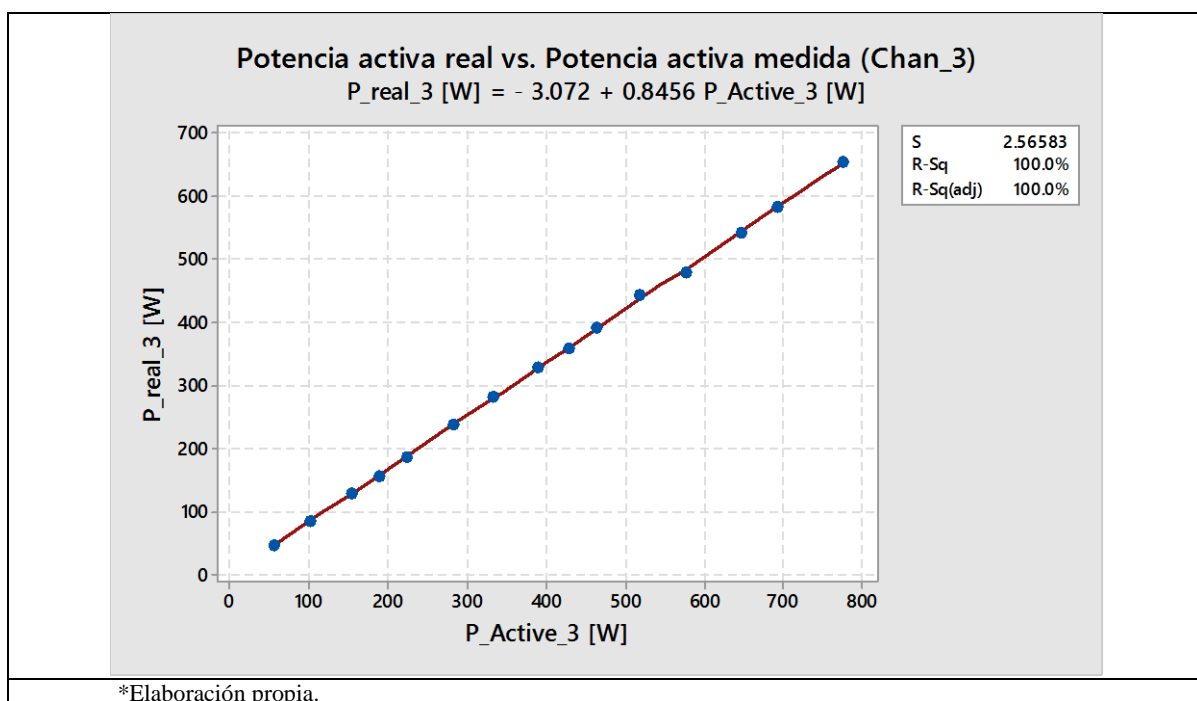


Figura 58: Gráfica de dispersión potencia activa Chan_3

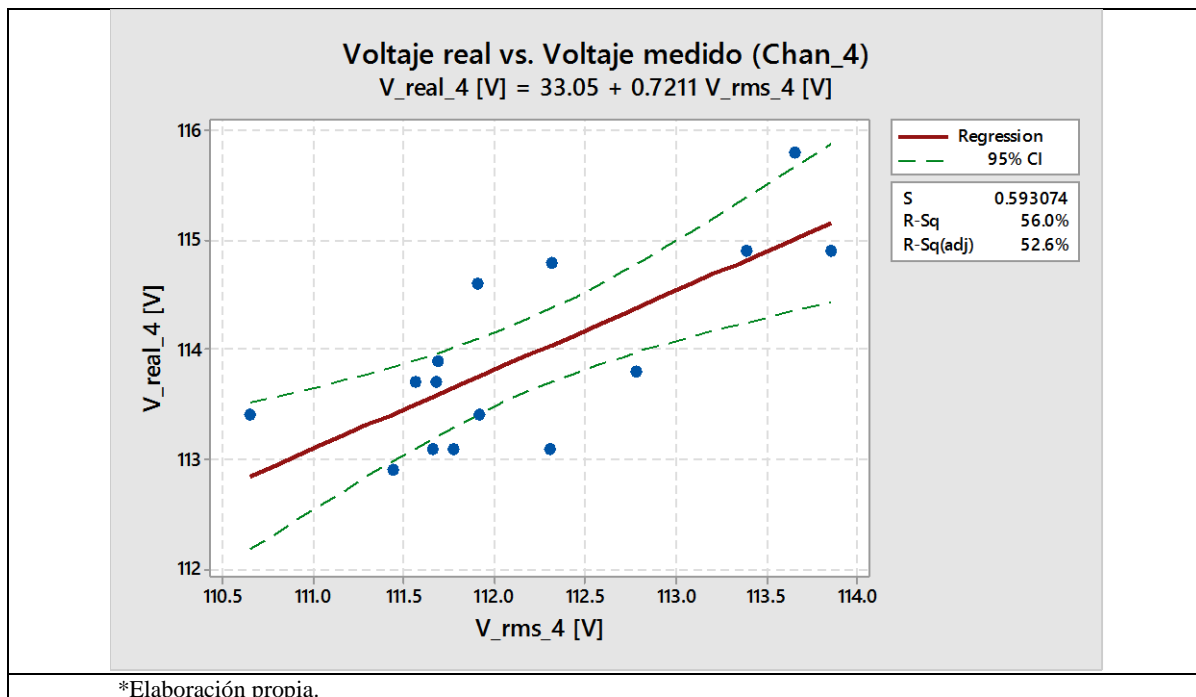


Figura 59: Gráfica de dispersión voltaje Chan_4

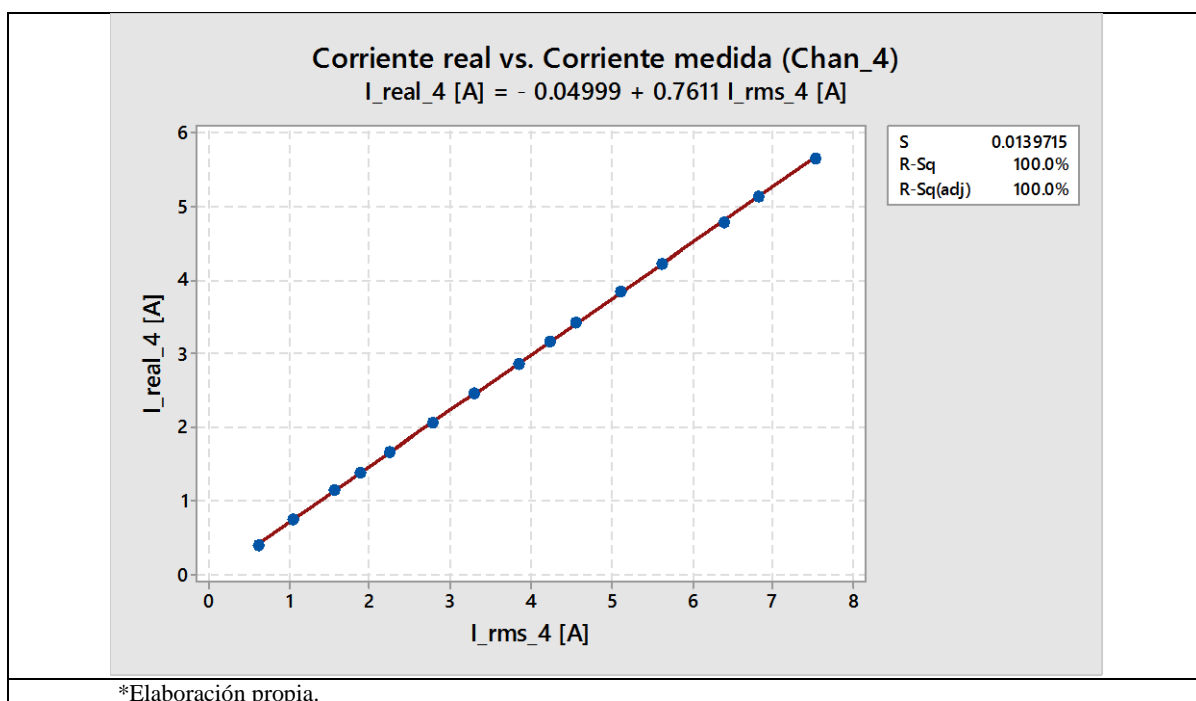


Figura 60: Gráfica de dispersión corriente Chan_4

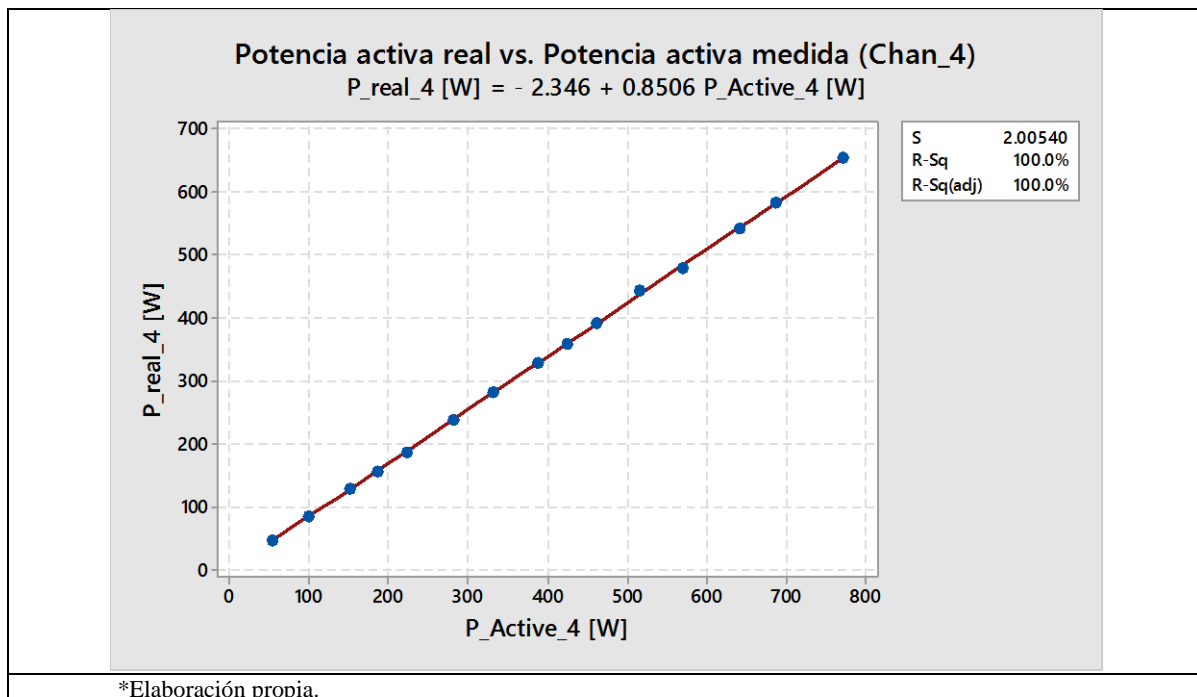


Figura 61: Gráfica de dispersión potencia activa Chan_4

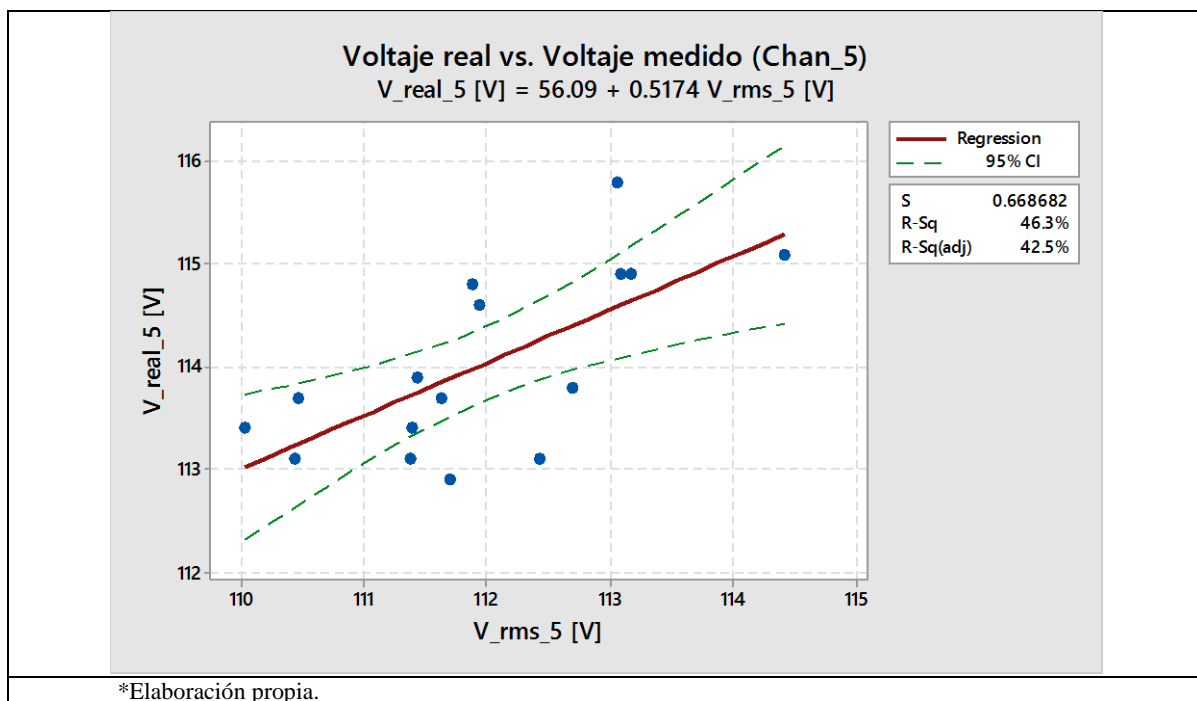


Figura 62: Gráfica de dispersión voltaje Chan_5

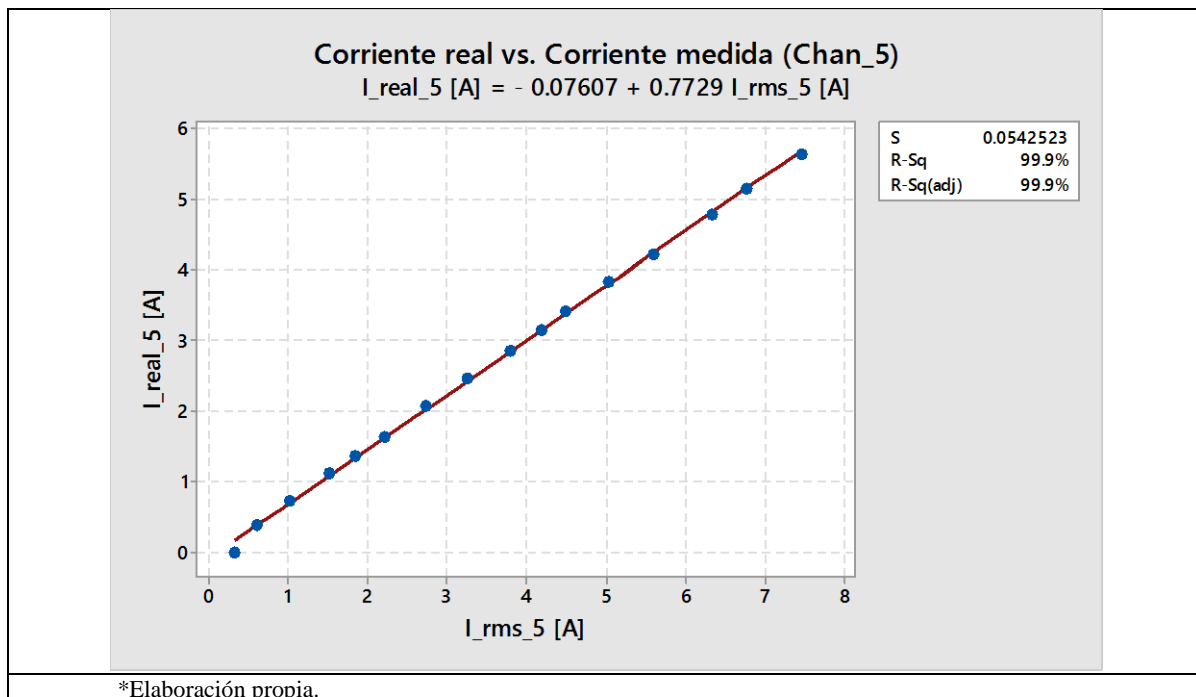


Figura 63: Gráfica de dispersión corriente Chan_5

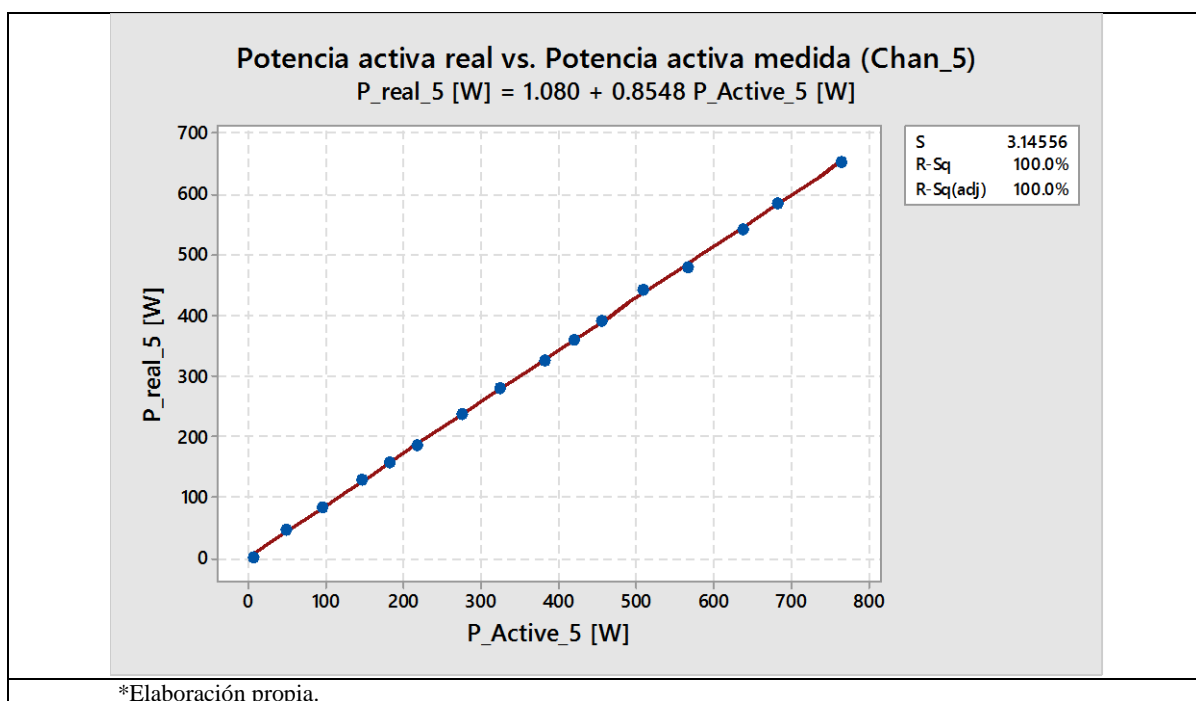


Figura 64: Gráfica de dispersión potencia activa Chan_5

ANEXO F 1.4: Mediciones por canal del sistema DAQ con funciones de corrección

Tabla 22: Mediciones Canal_2 con corrección (promedios)

V_real_2 [V]	V_rms_2 [V]	I_real_2 [A]	I_rms_2 [A]	P_real_2 [W]	P_Active_2 [W]	Factor_P_2	Frecuency_2 [Hz]
114.300	113.361	0.000	0.172	0.000	0.000	0.109	59.981
115.800	115.003	0.400	0.446	46.320	43.908	0.857	59.991
117.200	116.785	0.920	0.950	107.824	105.002	0.946	60.016
115.200	114.251	1.130	1.153	130.176	126.401	0.959	59.961
114.100	114.028	1.630	1.656	185.983	183.578	0.972	59.984
114.000	114.507	2.250	2.297	256.500	257.611	0.979	60.002
113.300	112.832	2.610	2.634	295.713	291.963	0.982	59.975
113.100	112.785	2.830	2.865	320.073	318.019	0.984	60.017
116.300	115.576	3.120	3.152	362.856	358.314	0.984	59.990
115.900	115.518	3.500	3.528	405.650	401.619	0.985	60.033
114.900	114.581	4.120	4.120	473.388	466.609	0.988	60.000
114.500	114.823	4.450	4.432	509.525	503.338	0.989	59.987
115.100	114.620	4.710	4.727	542.121	536.358	0.990	59.998
115.600	115.188	5.360	5.373	619.616	613.562	0.991	59.991
115.800	114.670	5.780	5.731	669.324	652.471	0.993	59.999

Tabla 23: Mediciones Canal_3 con corrección (promedios)

V_real_3 [V]	V_rms_3 [V]	I_real_3 [A]	I_rms_3 [A]	P_real_3 [W]	P_Active_3 [W]	Factor_P_3	Frecuency_3 [Hz]
114.300	113.356	0.000	0.200	22.700	0.000	0.100	59.967
115.800	115.368	0.400	0.455	52.490	43.937	0.837	59.992
117.200	117.022	0.920	0.957	111.960	105.340	0.941	60.013
115.200	114.683	1.130	1.151	132.012	125.817	0.953	59.972
114.100	113.931	1.630	1.647	187.623	181.862	0.969	59.991
114.000	114.645	2.250	2.284	261.869	255.996	0.978	60.000
113.300	113.135	2.610	2.621	296.534	290.898	0.981	59.965
113.100	112.954	2.830	2.853	322.231	316.641	0.983	60.008
116.300	115.488	3.120	3.130	361.506	354.827	0.981	59.988
115.900	115.522	3.500	3.500	404.370	397.516	0.983	60.016
114.900	114.890	4.120	4.112	472.409	465.808	0.986	59.995
114.500	115.008	4.450	4.412	507.411	500.832	0.987	59.986
115.100	115.073	4.710	4.719	543.033	536.432	0.988	60.000
115.600	115.610	5.360	5.366	620.379	613.552	0.989	59.997
115.800	115.208	5.780	5.727	659.816	653.226	0.990	59.990

Tabla 24: Mediciones Canal_4 con corrección (promedios)

V_real_4 [V]	V_rms_4 [V]	I_real_4 [A]	I_rms_4 [A]	P_real_4 [W]	P_Active_4 [W]	Factor_P_4	Frecuency_4 [Hz]
114.30	115.07	0.00	0.21	0.00	0.00	0.08	59.99
115.80	115.17	0.40	0.45	46.32	42.86	0.83	59.99
117.20	117.02	0.92	0.95	107.82	104.40	0.94	60.03
115.20	115.06	1.13	1.14	130.18	124.85	0.95	59.97
114.10	114.88	1.63	1.64	185.98	181.97	0.97	59.99
114.00	114.70	2.25	2.26	256.50	252.75	0.97	60.00
113.30	113.54	2.61	2.60	295.71	287.76	0.98	59.99
113.10	113.60	2.83	2.84	320.07	315.14	0.98	60.02
116.30	115.71	3.12	3.11	362.86	353.11	0.98	59.99
115.90	115.80	3.50	3.48	405.65	396.05	0.98	60.02
114.90	115.46	4.12	4.09	473.39	464.85	0.98	60.00
114.50	115.44	4.45	4.39	509.53	499.16	0.98	59.99
115.10	115.62	4.71	4.70	542.12	535.88	0.99	60.00
115.60	115.67	5.36	5.31	619.62	606.49	0.99	59.98
115.80	115.16	5.78	5.67	669.32	645.53	0.99	59.99

Tabla 25: Mediciones Canal_5 con corrección (promedios)

V_real_5 [V]	V_rms_5 [V]	I_real_5 [A]	I_rms_5 [A]	P_real_5 [W]	P_Active_5 [W]	Factor_P_5	Frecuency_5 [Hz]
114.300	113.717	0.000	0.253	0.000	0.000	0.041	59.989
115.800	114.436	0.400	0.461	46.320	41.904	0.795	59.991
117.200	115.491	0.920	0.957	107.824	102.813	0.931	60.017
115.200	114.223	1.130	1.153	130.176	123.865	0.940	59.965
114.100	113.917	1.630	1.662	185.983	180.618	0.954	59.988
114.000	114.233	2.250	2.305	256.500	253.424	0.962	60.009
113.300	113.315	2.610	2.647	295.713	288.119	0.961	59.968
113.100	113.117	2.830	2.883	320.073	313.139	0.960	60.017
116.300	114.585	3.120	3.157	362.856	350.152	0.968	59.994
115.900	114.805	3.500	3.550	405.650	395.317	0.970	60.026
114.900	114.383	4.120	4.165	473.388	462.296	0.970	59.997
114.500	114.519	4.450	4.483	509.525	498.794	0.971	59.983
115.100	114.648	4.710	4.799	542.121	535.214	0.973	60.003
115.600	114.574	5.360	5.423	619.616	605.075	0.974	59.992
115.800	114.441	5.780	5.814	669.324	647.948	0.974	59.993

ANEXO F 1.5: Tablas de error para las mediciones del sistema DAQ con funciones de corrección

Tabla 26: Tabla de error Canal_2

Set de datos	Error Voltaje	Error Corriente	Error P. Activa	Factor de Potencia
1	0.69%	11.42%	5.21%	14.35%
2	0.35%	3.28%	2.62%	5.41%
3	0.82%	2.06%	2.90%	4.08%
4	0.06%	1.57%	1.29%	2.77%
5	0.44%	2.07%	0.43%	2.06%
6	0.41%	0.94%	1.27%	1.81%
7	0.28%	1.24%	0.64%	1.61%
8	0.62%	1.01%	1.25%	1.65%
9	0.33%	0.79%	0.99%	1.47%
10	0.28%	0.00%	1.43%	1.16%
11	0.28%	0.40%	1.21%	1.13%
12	0.42%	0.36%	1.06%	1.02%
13	0.36%	0.25%	0.98%	0.88%
14	0.98%	0.85%	2.52%	0.72%
Error medio	0.45%	1.88%	1.70%	2.86%

Tabla 27: Tabla de error Canal_3

Set de datos	Error Voltaje	Error Corriente	Error P. Activa	Factor de Potencia
1	0.37%	13.74%	16.30%	16.31%
2	0.15%	3.99%	5.91%	5.92%
3	0.45%	1.86%	4.69%	4.69%
4	0.15%	1.02%	3.07%	3.07%
5	0.57%	1.51%	2.24%	2.25%
6	0.15%	0.42%	1.90%	1.92%
7	0.13%	0.80%	1.73%	1.74%
8	0.70%	0.32%	1.85%	1.86%
9	0.33%	0.00%	1.70%	1.71%
10	0.01%	0.20%	1.40%	1.39%
11	0.44%	0.86%	1.30%	1.29%
12	0.02%	0.18%	1.22%	1.23%
13	0.01%	0.11%	1.10%	1.11%
14	0.51%	0.93%	1.00%	1.01%
Error medio	0.28%	1.85%	3.24%	3.25%

Tabla 28: Tabla de error Canal_4

Set de datos	Error Voltaje	Error Corriente	Error P. Activa	Factor de Potencia
1	0.55%	12.67%	7.46%	17.43%
2	0.16%	3.10%	3.18%	5.94%
3	0.12%	0.98%	4.09%	4.92%
4	0.68%	0.67%	2.16%	3.48%
5	0.61%	0.58%	1.46%	2.65%
6	0.22%	0.43%	2.69%	2.49%
7	0.44%	0.33%	1.54%	2.31%
8	0.50%	0.19%	2.69%	2.03%
9	0.08%	0.46%	2.37%	1.84%
10	0.49%	0.66%	1.80%	1.64%
11	0.82%	1.30%	2.03%	1.57%
12	0.45%	0.20%	1.15%	1.42%
13	0.06%	0.92%	2.12%	1.29%
14	0.55%	1.87%	3.55%	1.18%
Error medio	0.41%	1.74%	2.74%	3.59%

Tabla 29: Tabla de error Canal_5

Set de datos	Error Voltaje	Error Corriente	Error P. Activa	Factor de Potencia
1	1.18%	15.19%	9.53%	20.53%
2	1.46%	3.97%	4.65%	6.95%
3	0.85%	2.06%	4.85%	5.98%
4	0.16%	1.97%	2.88%	4.62%
5	0.20%	2.45%	1.20%	3.76%
6	0.01%	1.40%	2.57%	3.93%
7	0.01%	1.88%	2.17%	3.99%
8	1.47%	1.18%	3.50%	3.22%
9	0.94%	1.42%	2.55%	3.01%
10	0.45%	1.10%	2.34%	2.97%
11	0.02%	0.74%	2.11%	2.87%
12	0.39%	1.88%	1.27%	2.72%
13	0.89%	1.18%	2.35%	2.65%
14	1.17%	0.59%	3.19%	2.64%
Error medio	0.66%	2.65%	3.23%	4.99%