

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA  
INDUSTRIAL

**Trabajo Fin de Grado**

DISEÑO DE UN E-METER CON DETECCIÓN DE EVENTOS  
PARA DESAGREGACIÓN DE ENERGÍA.

ESCUELA POLITECNICA  
**Autor:** Inés Morales Domínguez  
SUPERIOR  
**Tutor/es:** Jesús Ureña Ureña

UNIVERSIDAD DE ALCALÁ  
Escuela Politécnica Superior

**GRADO EN INGENIERIA ELECTRÓNICA Y AUTOMÁTICA  
INDUSTRIAL.**

Trabajo Fin de Grado

Diseño de un E-Meter con detección de eventos para desagregación de  
energía.

**Autor:** Inés Morales Domínguez.

**Tutor:** Jesús Ureña Ureña.

**TRIBUNAL:**

**Presidente:** Pedro Martín Sánchez

**Vocal 1º:** Francisco Javier Rodríguez Sánchez

**Vocal 2º:** Jesús Ureña Ureña

**FECHA:** 27/09/2019

*“Disfruta lo que haces y entonces harás tu mejor esfuerzo.”*

Katherine Johnson

## **AGRADECIMIENTOS.**

A mi tutor, Jesús Ureña; gracias por demostrarme que aún quedan docentes que disfrutan de su trabajo. Sin tu ayuda y tu paciencia esto no hubiera sido posible.

A Manuela, quien me enseñó el difícil camino del reaprendizaje.

A Charo y a Jesús, quienes siempre me han ofrecido su ayuda sin esperar nada a cambio. Gracias por todo vuestro cariño y apoyo.

A mis dos sorpresas, Silvia y Nerea: Volvisteis a mi vida y espero que sea para siempre. Gracias por preguntar, por estar pendientes, por aguantar mis miedos y sobre todo por quedaros.

A Pilar, dicen que quienes son amigas durante 10 años ya lo son para siempre. Gracias por estar hasta cuando no nos aguantábamos. Te quiero.

A mi gran familia de la universidad: Díaz, Gallego, López, Álvaro, Ángulo, Pol y Cebri. Gracias por tantos momentos en la biblioteca, por tantos apuntes compartidos, por tantas fiestas y risas y por tantos lloros y ánimos. Sin vosotros esto no hubiera sido igual.

A Ainoa, mi gran amiga de la universidad. Gracias por todas las prácticas que hemos hecho juntas, por tus ánimos, por todos los lloros que has aguantado, las risas y todas las confidencias. Por tus “Inés lo resolveremos juntas como hemos hecho siempre”.

A mi gran compi de laboratorio, Rubén. Gracias por compartir conmigo el final de este camino y hacerme las cosas fáciles. No desesperes que el final está cerca compi.

A mi abuela, toda una superviviente. Gracias por recordar cada una de mis fechas de exámenes, gracias por hacer de mis logros los tuyos y sentirte orgullosa de mí.

A la mujer más valiente y con más fuerza de voluntad que he conocido, mi madre. Gracias por todas tus broncas, por todos tus consejos, aunque no te hiciera caso; pero sobre todo gracias por enseñarme a no tirar la toalla, a luchar por mis sueños. Si esto ha sido posible es en gran parte gracias a ti.

A Juan, lo mejor que me llevo de toda esta etapa. Contigo aprendí que quien no arriesga no gana. Gracias por aparecer en el peor momento de mi vida y decidir quedarte y aguantar, sé que no ha sido fácil. Eres todo un ejemplo para mí, gracias por confiar en mi cuando ni si quiera yo lo hacía. Te quiero.

A mi padre, el gran hombre de mi vida. No tengo palabras para expresar todo lo que siento por ti, mi mayor ejemplo. Gracias por entenderme mejor que nadie, por escucharme y por tener las palabras adecuadas en cada momento. Gracias por saber cómo aliviar mis nervios y mi ansiedad y por ser mi gran apoyo Papá. Gracias por confiar siempre e incondicionalmente en mí, por fin podemos decir que soy Ingeniera.

Y por último a mi abuelo. Sé que allá donde estés estás compartiendo este momento conmigo. Gracias por no soltarme nunca, por toda tu ayuda y por todo lo que he podido aprender contigo. Ojalá pudiera coger tu mano otra vez. Gracias, por tanto, te quiero yayo.

## ÍNDICE

1.	RESUMEN .....	7
2.	SUMMARY .....	7
3.	PALABRAS CLAVE.....	8
4.	RESUMEN EXTENDIDO .....	9
5.	MEMORIA.....	11
5.1	INTRODUCCIÓN.....	11
5.2	DESAGREGACIÓN DE ENERGÍA.....	13
5.3	MONITORIZACIÓN NO INTRUSIVA DE CARGAS (NILM) .....	14
5.3.1	ALGORITMOS NILM .....	16
5.4	METERING CON DETECCIÓN DE EVENTOS A ALTA FRECUENCIA DE MUESTREO.....	17
5.4.1	DETECCIÓN DE EVENTOS.....	18
5.4.2	EXTRACCIÓN DE LA SEÑAL DE LA CARGA .....	18
5.4.3	IDENTIFICACIÓN DE LA CARGA.....	21
5.5	SISTEMAS COMERCIALES: CONTADORES Y MEDIDORES EN EL MERCADO. ....	21
5.5.1	PROYECTO STAR DE IBERDROLA DISTRIBUCIÓN .....	22
5.5.2	MIRUBEE.....	23
5.5.3	INFORMETIS + TEPCO.....	26
5.5.4	HOWZ.....	27
5.5.5	ENERGENIE: MI HOME ECO PACK.....	27
5.5.6	PEACEFAIR .....	28
5.5.7	WIBEEE.....	28
5.6	HARDWARE .....	30
5.6.1	TARJETA DE ADQUISICIÓN “EV-ADE9153ASHIELDZ”.....	31
5.6.2	CHIP AD9153A .....	32
5.6.2.1	Autocalibración: mSure .....	33
5.6.2.2	Convertor analógico-digital: ADC.....	33
5.6.2.3	Serial Peripheral Interface: SPI .....	35
5.6.2.4	Universal Asynchronous Receiver-Transmitter: UART.....	36
5.6.2.5	Interrupciones .....	36
5.6.2.6	Medidas.....	37
5.7	SOFTWARE Y DATOS .....	39
5.7.1	SOFTWARE EMPLEADO.....	39

5.7.1.1	Librerías .....	40
5.7.1.2	Código fuente. ....	42
5.7.1.3	Programa principal .....	44
5.7.2	DATOS .....	46
5.7.3	ALGORITMO .....	48
5.7.4	RESULTADOS. ....	51
5.8	CONCLUSIONES Y TRABAJOS FUTUROS .....	61
6.	CÓDIGO Y ALGORITMO .....	62
6.1	CÓDIGO PARA LA ADQUISICIÓN DE DATOS. ....	62
6.1.1	ADE9153AAPI.cpp .....	62
6.1.2	ADE9153AAPI_Test.ino .....	69
6.1.3	ALGORITMO .....	72
7.	PLIEGO DE CONDICIONES .....	74
7.1	CARACTERÍSTICAS DEL CHIP ADE9153 .....	74
7.2	DIAGRAMA DE CONEXIONES DE LA TARJETA.....	80
8.	PRESUPUESTO .....	81
9.	MANUAL DE USUARIO.....	82
9.1	HARDWARE Y SOFTWARE NECESARIO .....	82
9.2	MONTAJE HARDWARE .....	82
9.3	SOFTWARE.....	83
9.4	CAPTURA DE DATOS .....	86
10.	BIBLIOGRAFIA .....	88
	APENDICE 1 .....	90

## LISTA DE FIGURAS

Figura 1: Esquema del sistema global.....	13
Figura 2: Señal de consumo característica de una lavadora.....	14
Figura 3: Número de artículos publicados por año.....	15
Figura 4: Curvas v-i para 6 cargas diferentes [12] .....	20
Figura 5 . Instalación final de Mirubox mono con pinza amperimétrica adicional. [9] .	24
Figura 6: Hábitos de consumo de una vivienda.....	25
Figura 7: Diagrama de bloques del diseño HW de un E-Meter .....	30
Figura 8: Tarjeta de adquisición EV-ADE9153ASHIELDZ .....	31
Figura 9: Diagrama de bloques ADE9153A [13].....	32
Figura 10: Diagrama de bloques de la conversión A-D.....	33
Figura 11: Ejemplo de las operaciones de escritura y lectura en el SPI [13] .....	35
Figura 12: Cálculos de energía y potencia a partir de los valores instantáneos [13].....	38
Figura 13: Cálculo de las ganancias de tensión y corriente. ....	43
Figura 14: Función programada para empezar la calibración del canal de corriente .....	43
Figura 15: Flujograma programa principal: Inicialización parámetros. ....	44
Figura 16: Flujograma programa principal: Main loop .....	45
Figura 17: Interfaz Realterm .....	47
Figura 18: Datos relacionados a la conexión y desconexión de varias cargas .....	48
Figura 19: Diagrama de bloques del algoritmo desarrollado .....	50
Figura 20: Tensión y corrientes eficaces de una bombilla de 40 W .....	51
Figura 21: Potencias activa, reactiva y aparente de una bombilla de 40W .....	52
Figura 22:Eventos de una bombilla de 40W .....	52
Figura 23:Tensión y corriente eficaces de un secador de 2500 W .....	53
Figura 24: Potencias activa, reactiva y aparente de un secador de 2000 W.....	54
Figura 25: Eventos de un secador de 2500 W .....	55
Figura 26:Tensión y corriente eficaces de una carga no lineal .....	56
Figura 27: Potencias activa, reactiva y aparente de una carga no lineal. ....	57
Figura 28: Eventos de una carga no lineal. ....	58
Figura 29: Tensión y corriente eficaz de varias cargas.....	59
Figura 30: Potencias activa, reactiva y aparente de varias cargas .....	60
Figura 31: Eventos de varias cargas.....	60
Figura 32: Conexionado entre la tarjeta, la carga y la toma de alimentación .....	82
Figura 33: Tarjeta con módulo Arduino conectado.....	83
Figura 34: Incluir librería en el proyecto. ....	84
Figura 35: Cuadro de dialogo para incluir las librerías al proyecto .....	84
Figura 36: Librerías a incluir .....	85
Figura 37: Configuración de la UART en el programa Realterm .....	86
Figura 38: Pestaña capture del programa Realterm .....	87

## LISTA DE TABLAS

Tabla 1: Relación de pines entre ADE9153A y Arduino.....	32
Tabla 2: Registros relacionados con la corriente en el ADE9153A .....	37
Tabla 3: Registros relacionados con la tensión en el ADE9153A.....	37
Tabla 4: Registros relacionados con la energía y la potencia en el ADE9153A.....	38
Tabla 5: Coste equipo empleado.....	81
Tabla 6: Coste software empleado.....	81
Tabla 7: Coste de personal .....	81
Tabla 8: Coste total del proyecto .....	81



## **1. RESUMEN**

En el presente TFG pretende diseñar un sistema de adquisición de variables eléctricas con una frecuencia de muestro suficientemente elevada como para programar algoritmos de detección de eventos para una posterior aplicación como puede ser la desagregación de energía y con ello la monitorización de personas a tiempo real. Como solución se propone un diseño hardware basado en el chip ADE9153 para extraer los datos de la tensión y corriente eficaces, consiguiendo una frecuencia de adquisición de 4 kHz. A su vez los datos obtenidos se procesarán, y se aplicará un algoritmo capaz de detectar las transiciones bruscas de nivel, identificando así la conexión y desconexión de algunos tipos de cargas a la red.

## **2. SUMMARY**

This TFG aims to design a system for acquiring electrical variables with a high sampling frequency to program event detection algorithms for a subsequent application such as energy disaggregation and ulterior real-time people monitoring. A hardware solution based on the ADE9153 chip is proposed to extract the voltage and current RMS data, obtaining an acquisition frequency of 4 kHz. The data obtained will be processed, and an algorithm capable of detecting abrupt level transitions will be applied, identifying the connection and disconnection of some loads to the network.

### **3. PALABRAS CLAVE**

- 1. Smart Meters.**
- 2. Smart Grids**
- 3. Desagregación de energía.**
- 4. Non-Intrusive Load Monitoring.**

## 4. RESUMEN EXTENDIDO

Actualmente, uno de los retos más importantes de nuestra sociedad es la distribución y el uso eficiente de la energía, surgiendo así la idea de Smart Grids (Redes Inteligentes de Energía). En ellas, los medidores inteligentes de energía (E-Meters) o dispositivos similares diseñados ad-hoc, conectados a la entrada de la instalación eléctrica, son capaces de funcionar como un único sensor. Dichos contadores inteligentes suministran, como elemento básico, los datos de consumo y pueden llegar a realizar ciertas estimaciones de consumo o dar el consumo en tiempo real de la instalación.

A su vez, surgen nuevas ideas de cara a la utilidad de los E-Meters; como por ejemplo la monitorización de personas analizando los distintos hábitos de consumo mediante la programación de ciertos algoritmos sobre los datos de la energía desagregada. Sin embargo, en la mayoría de los medidores actuales que se encuentran en el mercado la desagregación de energía no es a tiempo real [1], lo que supone un inconveniente a la hora de poder realizar la monitorización de personas a partir de sus hábitos de consumo.

En este sentido, este TFG se centra en diseñar un sistema de adquisición de variables eléctricas con una frecuencia de muestreo del orden de kilohercios para programar algoritmos de detección de eventos para una posterior aplicación en la desagregación de energía. Así mediante la detección de la conexión y desconexión de cargas a la red eléctrica de una instalación, la desagregación de energía puede ser a tiempo real y, por lo tanto, se podría llegar a monitorizar a una persona teniendo en cuenta sus hábitos de consumo o patrones previamente determinados de consumo diarios y dicha desagregación de energía.

Este proyecto, por tanto, tiene dos partes diferenciadas: La adquisición de datos y la aplicación del algoritmo de detección de eventos sobre estos datos. Para la adquisición de variables eléctricas se han aprovechado las prestaciones del e-meter ADE9153A de Analog Devices. Este dispositivo adquiere la corriente y la tensión instantáneas a una frecuencia de adquisición de 4 kHz. A partir de ellas calcula el resto de los parámetros: Corriente y tensión eficaces, valores de energías y potencias y factor de potencia. Este chip está integrado en la tarjeta "EV-ADE9153ASHIELDZ", la cual permite conectar de forma directa un módulo Arduino y procesar los datos con el mismo. Debido a esta prestación se ha optado por emplear este microprocesador para el procesado de los datos obtenidos.

Para la captura de datos se han realizado pruebas con tres tipos de carga distintos, es decir una carga resistiva, otra de tipo inductivo y otra capacitivo. Se han obtenido datos relacionados con los valores eficaces, las potencias y el factor de potencia.

Estos datos se han cargado en Matlab y se han obtenido todas las gráficas relacionadas con las mismas para comprobar el funcionamiento de cada carga. También se han obtenido y procesado datos relacionados a la conexión y desconexión de estas tres cargas a la vez.

Una vez que se han obtenido las distintas gráficas se ha aplicado el algoritmo de detección de eventos propuestos a la señal de corriente eficaz, ya que la corriente es la que más información alberga acerca de la carga debido a que es especialmente sensible a las distintas conexiones y desconexiones que tienen lugar en la red.

El algoritmo se basa principalmente, en detectar las transiciones bruscas que tienen lugar en la señal, que indican las conexiones y desconexiones que tienen lugar en la red eléctrica de la instalación o los cambios que pueda haber en ésta y etiquetarlos representando los resultados del algoritmo.

En cuanto a la memoria del TFG se ha estructurado en los siguientes apartados:

1. Memoria: Este es el bloque principal en el cual se recoge el estado del arte relacionado con la desagregación de energía. En los primeros apartados siguientes se recogerá el estado del arte relacionado con la desagregación de energía y los sistemas de monitorización de carga no intrusiva. También se describe la tarjeta de adquisición empleada y toda la fase experimental del presente proyecto recogiendo los resultados y su explicación. Al final de este bloque se incluye un apartado de conclusiones y trabajos futuros.
2. Código: En este apartado se han incluido los códigos que han sido empleados para la adquisición de datos y el código que se ha desarrollado para el algoritmo.
3. Pliego de condiciones: Se recogen las hojas de características del chip empleado y el diagrama de conexiones de la tarjeta.
4. Presupuesto: En este apartado se recoge el coste relacionado con el material empleado y la mano de obra necesaria para la realización de este proyecto, así como el coste total.
5. Manual de usuario: Se incorpora un manual de usuario en el que se describen los pasos que hay que seguir para la adquisición y procesado de los datos.
6. Bibliografía con las fuentes consultadas.

Apéndice: En el apéndice se recogen los informes de consumo de una vivienda en la que se ha instalado el e-meter de Mirubee.

## 5. MEMORIA

### 5.1 INTRODUCCIÓN.

En este apartado se pretende exponer los objetivos de este TFG y dar una explicación general y breve a cerca del diseño hardware, el algoritmo propuestos y el material utilizado.

El presente proyecto se centra en el estudio y prueba de un diseño hardware de un E-Meter que sea capaz de extraer los datos de consumo de una instalación eléctrica a suficiente frecuencia como para, posteriormente, poder implementar algoritmos de detección de eventos (principalmente apagado/encendido de cargas eléctrica). Sobre estos datos se podría en trabajos futuros conseguir la desagregación de consumo de energía en tiempo real y enfocarla a la monitorización de personas.

Para conseguir que la desagregación de energía se pueda realizar en tiempo real, el algoritmo que se propone debe detectar los eventos y añadirles una etiqueta temporal; es decir, la detección del momento de la conexión y desconexión de cargas a la red eléctrica. Para ello se extraen los datos de corriente y de tensión y con ellos se detectan cambios significativos en sus valores eficaces (sobre todo de la corriente). A partir del instante en que se detecta el evento, se abre una ventana para calcular las distintas potencias (activa, reactiva y de distorsión). Posteriormente, estos datos se confrontarían con bases de datos de huellas de consumo de las distintas cargas en el hogar para realizar un reconocimiento de patrones que concluye con la asignación de cada evento a la conexión o desconexión de un tipo de carga concreto.

Es importante señalar que para que el algoritmo propuesto sea eficiente se necesita una frecuencia de muestreo lo suficientemente elevada, del orden de algunos kilohercios; ya que lo que se pretende es detectar los eventos que hay en una señal (transiciones bruscas) así como, los transitorios que se producen debido a esas transiciones. De esta forma, cuanto mayor sea la frecuencia de muestreo mayor será el número de muestras y la capacidad de resolución.

Después de realizar un estudio de los distintos chips específicos para e-metering (con sistema de adquisición y análisis de potencias) que están disponibles en el mercado, se ha intentado escoger aquél que permitiera una frecuencia de muestreo lo más elevada posible. De esta manera, se ha optado por aprovechar las prestaciones de un E-Meter basado en el chip ADE9153A de Analog Devices.

La tarjeta de evaluación en la que está integrado el chip es la “EV-ADE9153ASHIELDZ”. Ésta se divide en dos partes:

- La primera de ellas obtiene los datos de la tensión y la corriente conectándola directamente a la instalación eléctrica, por lo que se trabaja en el rango de tensiones habituales de baja tensión (por ejemplo, 220 V).
- La segunda de ellas es la que permite trabajar directamente con los datos de corriente y tensión de una forma segura ya que funciona a 5 V. Ambas partes están totalmente aisladas mediante un módulo optoacoplador.

Para poder configurar la tarjeta y procesar los datos se emplea un módulo de Arduino Uno. Éste se conecta a la tarjeta de adquisición mediante un puerto SPI, consiguiéndose unos 76 datos/s, debido a la velocidad de funcionamiento del SPI.

Por último, se realizan simulaciones con Matlab de los datos adquiridos. En estas simulaciones se aplican técnicas de detección de eventos pudiendo así etiquetar las conexiones/desconexiones de las distintas cargas que se encuentran conectadas a la red.

La estructura de la presente memoria se puede dividir en 4 grandes bloques:

1. En los primeros apartados siguientes se recogerá el estado del arte relacionado con la desagregación de energía, los sistemas de monitorización de carga no intrusiva y se realizará una breve explicación de los distintos Smart-Meters y dispositivos ad-hoc para desagregación de energía que podemos encontrar en el mercado actual.
2. Posteriormente se describirá tanto la tarjeta de adquisición empleada y las distintas variables del chip que se pueden encontrar, como el diseño hardware propuesto.
3. En los siguientes apartados se explicarán las distintas técnicas y algoritmos que se han empleado para procesar y simular los datos y realizar la detección de eventos.
4. Por último, se recogerán los resultados de las diferentes simulaciones realizadas junto con las conclusiones obtenidas. También habrá un apartado en el que se proponen los futuros trabajos y mejoras relacionadas con el presente TFG.

## 5.2 DESAGREGACIÓN DE ENERGÍA

La desagregación de energía es un conjunto de técnicas y algoritmos mediante los cuales se pueden reconocer las distintas cargas que se encuentran conectadas a la red eléctrica, a partir de la potencia total consumida.

Para poder determinar cuál es la potencia consumida por cada electrodoméstico o aparato electrónico conectado a la red eléctrica es necesario obtener la corriente y la tensión y posteriormente calcular las potencias activa, reactiva y de distorsión.

Así, para poder obtener todos estos datos es necesario el uso de un contador o medidor inteligente que se encuentre conectado a la red eléctrica y del cual se puedan obtener los datos necesarios. La figura 1 muestra un esquema del sistema global.

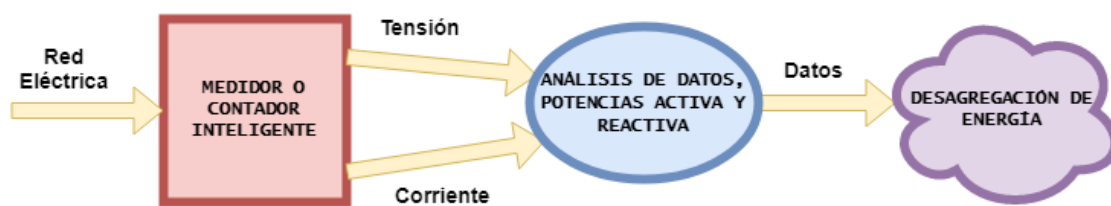


Figura 1: Esquema del sistema global.

Entre las técnicas empleadas en la desagregación de energía se pueden señalar varias, destacando técnicas de procesamiento de señales, reconocimiento de patrones y machine learning.

Este TFG se centra en la determinación de eventos de conexión/desconexión de cargas, de modo que posteriormente se pueda realizar un reconocimiento de patrones. Esta técnica, puede ser, por ejemplo, el analizar los datos de las potencias activa, reactiva y de distorsión y las características particulares de cada carga que se encuentra conectada a la red eléctrica en el estado transitorio de la señal.

Así, por ejemplo; el frigorífico tiene una señal característica con picos de consumo cada vez que arranca el compresor y la señal de la lavadora tiene unos ‘zig-zags’ cada vez que para y arranca el motor que da vueltas al tambor. [1], la cual se puede ver en la figura 2 [21].

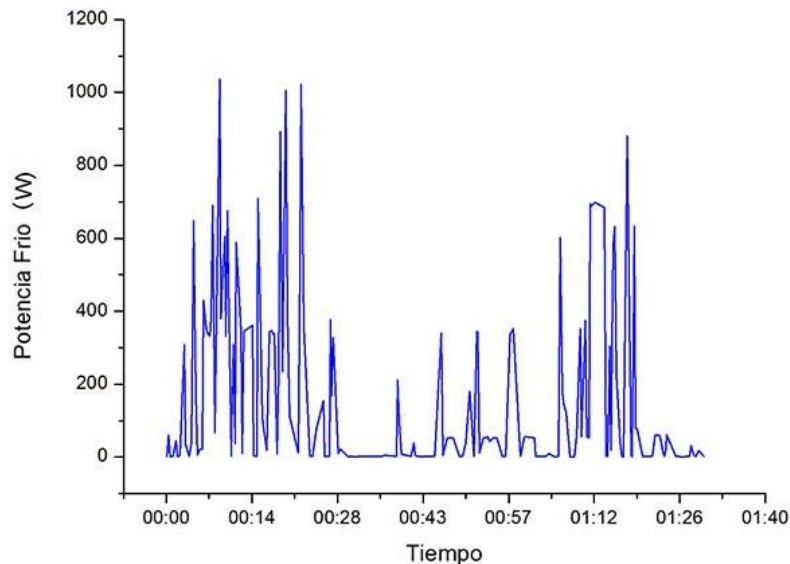


Figura 2: Señal de consumo característica de una lavadora.

### 5.3 MONITORIZACIÓN NO INTRUSIVA DE CARGAS (NILM)

Dentro de la desagregación de energía hay que destacar los sistemas de monitorización de carga no intrusiva (NILM- *Non-Intrusive Load Monitoring*).

Los sistemas NILM tratan de predecir qué dispositivos hay conectados a la red eléctrica y qué están consumiendo a partir de un único sensor: el contador inteligente o un dispositivo ad-hoc suministrado por terceras partes (no necesariamente suministradores de contadores). Muchas veces, los datos son obtenidos por el medidor y accesibles remotamente a través de la propia red “*Power Line Communication*” (PLC) [2].

Los contadores inteligentes evolucionaron de cara a conseguir una mayor eficiencia energética debido a la preocupación acerca de la escasez de los recursos energéticos y el cambio climático.

Así, los sistemas NILM nacen gracias al avance de los contadores inteligentes, aunque los primeros trabajos se remontan a 1992: nacen las aplicaciones de monitorización no intrusiva de cargas de la mano de George Hart. Ahí se sentaron las bases de muchos de los conceptos de los NILM [2].



Actualmente, el interés por estos sistemas está en aumento. Así, en la figura 3 se puede observar cómo el número de artículos publicados relacionados con la desagregación de energía y NILM han aumentado de forma exponencial desde el 2010 [4].

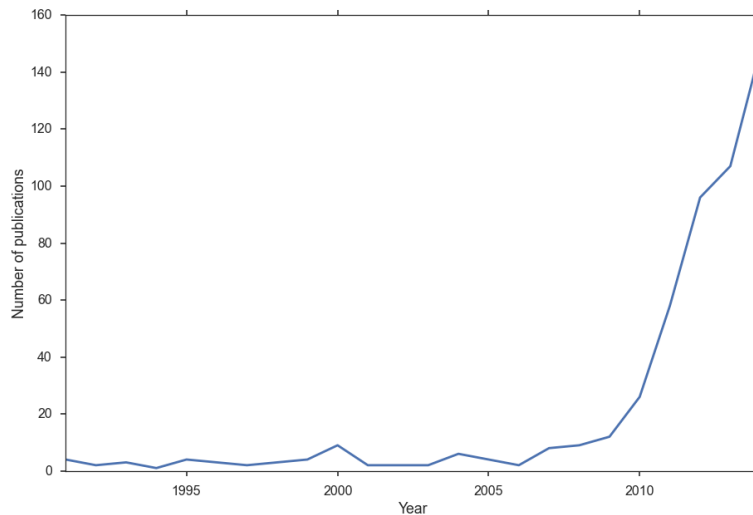


Figura 3: Número de artículos publicados por año

Esta figura demuestra el interés de la comunidad científica en torno a los sistemas NILM. Así, las grandes compañías eléctricas cada vez apuestan más por incluir esta tecnología en sus negocios, sin embargo, todavía sigue estando destinada al consumidor final.

Aparte de desarrollar esta tecnología para su uso en múltiples aplicaciones, también se trabaja para mejorar algunos aspectos de la misma como, por ejemplo, conseguir una alta fiabilidad en la extracción y análisis de los datos en tiempo real. Actualmente la mayoría de los *Smart meters* que incorporan esta tecnología no trabajan en tiempo real, en el sentido de dar resultados con pequeña latencia, ya que sus algoritmos se basan en la acumulación de datos extraídos por el sensor y análisis de los mismos en un tiempo posterior.

Algunas de las aplicaciones que tienen hoy en día son: [3]

- Identificación de carga “extremadamente rentable”.
- Detección barata de transitorios de arranque, fallas de línea o equipo.
- Monitorización de los sistemas de a bordo de los barcos, consiguiendo que éstos operen de forma más segura a través de un número más reducido de sensores.
- Monitorización de personas a través de sus hábitos de consumo doméstico.

### 5.3.1 ALGORITMOS NILM

Los algoritmos y métodos empleados en los sistemas de monitorización de carga no intrusiva se pueden englobar en 3 grandes grupos:

1. Algoritmos de clasificación por evento o por no evento.
2. Algoritmos supervisados y no supervisados.
3. Algoritmos basados en la frecuencia de muestreo.

A continuación, se explican brevemente cada uno de ellos y sus diferencias.

- **Algoritmos de clasificación por evento o por no evento:** Los algoritmos por detección de eventos tratan de detectar y clasificar los estados transitorios, detectando, por tanto, cambios de consumo o cambios en el estado de las cargas. Los métodos basados en no eventos no necesitan detectar cambios de consumo ya que, a partir de un número de observaciones prevén el estado de un conjunto de cargas en cada muestra.
- **Algoritmos supervisados, no supervisados y semi-supervisados:** Estos métodos se basan en técnicas de *machine learning*. Así, los algoritmos supervisados entrenan al modelo usando datos etiquetados. En este tipo de técnica se necesita de un monitoreo intrusivo.  
Por otro lado, los algoritmos no supervisados difieren un poco en cuanto a las técnicas no supervisadas puras de *machine learning*. En este caso los métodos no supervisados necesitan para entrenar al modelo datos no etiquetados, sin embargo, en NILM, se usan datos no etiquetados y etiquetados de otros modelos, por ejemplo; una instalación eléctrica de un domicilio de prueba.
- **Algoritmos basados en la frecuencia de muestreo:** En este caso, los algoritmos se enfocan en detectar eventos y se distinguen dos tipos; métodos de baja frecuencia y métodos de alta frecuencia.  
Antes de analizar cada uno de los casos cabe resaltar que, tanto la integración, como la desagregación de la energía, se encuentran estrechamente relacionados con la frecuencia de muestreo. De esta manera la precisión de la desagregación incrementa de forma proporcional con la frecuencia de muestreo mientras que la integración decrece.  
En cuanto a los métodos basados en baja frecuencia, éstos trabajan con una frecuencia de muestreo comprendida en el rango de 0,1 Hz y 1 Hz más o menos, por lo que tan sólo se puede extraer la información del régimen permanente. Así los datos que se pueden extraer son datos relacionados con la potencia activa, como, por ejemplo; el valor de ésta en distintos estados estacionarios y el incremento de esta entre dos estados permanentes. Con esta resolución no se puede llegar a obtener información de los armónicos de la señal. Este tipo de método se basa en la predicción de los estados de las cargas para cada tiempo de muestreo. Así, el número de interacciones incrementa conforme lo hace el número de muestras. Actualmente, el procesamiento de los datos obtenidos bajo este algoritmo se realiza bajo el enfoque del modelo factorial oculto de Markov. [2]  
Con una frecuencia de muestreo algo mayor, comprendida entre 1 Hz y 20 KHz, se puede extraer mucha más información: los armónicos de la señal, las formas de

las ondas y las potencias activa, reactiva y de distorsión. También se pueden llegar a analizar los transitorios de las distintas señales y se obtiene una precisión adecuada.

Por último, con una frecuencia de muestreo elevada, mayor de 100 KHz se puede llegar a obtener información del ruido de la señal, captando las interferencias electromagnéticas (EMI), tanto su estado permanente como el transitorio.

Para la extracción de los datos y su posterior análisis, el presente proyecto se basa en la detección de eventos con una frecuencia de muestreo media, suficiente para la detección de eventos. Los algoritmos se explican con más profundidad en el apartado 6.4

## 5.4 METERING CON DETECCIÓN DE EVENTOS A ALTA FRECUENCIA DE MUESTREO.

En este apartado se explican los algoritmos basados en la detección de eventos para frecuencias bastante mayores a 1 Hz.

Los algoritmos basados en la detección eventos a baja frecuencia tienen algunas ventajas como que la desagregación de energía es sesgada y los estados son mucho más fáciles de predecir; sin embargo, no reflejan la frecuencia de la actividad humana, de ahí la necesidad de aplicar los algoritmos basados en alta frecuencia de muestreo ya que permiten etiquetar eventos y aumenta la precisión en la desagregación de energía. A su vez, la integración de estos algoritmos (basados en detección a baja frecuencia) en los *Smart meters* es peor que la integración de algoritmos que no están basados en la detección de eventos, sin embargo, son menos dependientes con respecto a la demanda de potencia de la aplicación [2].

Los algoritmos basados en detección de eventos normalmente tienen 3 etapas: la propia detección de eventos, la extracción de la señal de la carga y la identificación de ésta.

Antes de aplicar la detección de eventos y el proceso de identificación de la carga las señales suelen ser normalizadas. El proceso de normalización es un proceso de aproximación de la señal de gran ayuda en el preprocesamiento de los datos ya que generaliza los cálculos para cualquier situación. En este caso, la señal de corriente es muy sensible a los cambios introducidos por las distintas cargas que se conectan a la red, las cuales introducen saltos de tensión, ruidos y distorsión. Así para normalizar la señal de la potencia se aplica la siguiente fórmula:

$$P_{norm}(t) = \left( \frac{V_{norm}}{v(t)} \right)^2 * p(t)$$

*Ecuación 1: Potencia normalizada.*

Donde:

- $V_{norm} = 230 V_{rms}$ .
- $p(t)$  y  $v(t)$  representan las medidas de potencia y tensión desde la red.

### 5.4.1 DETECCIÓN DE EVENTOS.

El principal objetivo de la detección de eventos es detectar los estados transitorios en la potencia producidos por la conexión y desconexión de cargas a la red. Detectando estos cambios se puede conocer dónde se ha producido una nueva conexión y/o desconexión, extraer la señal propia de la carga y conocer sus características.

Es común que los métodos de detección de eventos se usen en las señales de corriente o de potencia ya que la tensión apenas varía con el tiempo, aunque la conexión de cargas introduce ruido en el voltaje en el espectro de 10 kHz a 100 kHz [2].

Dentro de la detección de eventos se pueden distinguir los siguientes métodos: Métodos heurísticos, modelos de probabilidad y filtros emparejados [2]. A continuación, se explica de forma breve en que se basa cada uno de ellos.

- **Métodos heurísticos:** Están basados en que se tiene cierto conocimiento sobre las cargas que se van a estudiar, ya que en numerosas ocasiones se necesita inicializar algunos parámetros. Algunos de estos parámetros pueden ser el número de datos necesarios en régimen transitorio o la potencia total demandada por la carga.
- **Modelos de probabilidad:** Estos métodos asignan una probabilidad que es usada por el algoritmo para tomar decisiones cuando ocurren los eventos. Es por esto, que precisa de un entrenamiento para inicializar algunos parámetros y aprender los modelos de distintos escenarios [2].
- **Filtros emparejados:** Estos filtros no necesitan de un conocimiento previo acerca de las cargas a estudiar, sin embargo, necesitan de un rango mayor de datos para extraer la forma de la señal de las distintas cargas y relacionarla con algún patrón similar. En este caso se suelen normalizar las señales. Para detectar eventos, es común extraer la envolvente de la señal de corriente y emplear desde filtro sencillos, como filtros de promedio o de paso bajo, hasta filtros de Kalman.

Con el auge de la inteligencia artificial, las técnicas de *machine learning* y el análisis en el dominio de la frecuencia también son usadas para la detección de eventos y el reconocimiento de patrones en señales eléctricas, consiguiendo unos resultados bastante óptimos.

### 5.4.2 EXTRACCIÓN DE LA SEÑAL DE LA CARGA

Cada aplicación tiene una señal de carga única y puede ser extraída tanto en el régimen permanente como el transitorio, aunque éste último alberga mucha más información y de cara a la detección de eventos resulta mucho más interesante. Así se pueden dividir los estados transitorios de las cargas en 4 tipos:

1. Cargas de encendido/apagado.
2. Cargas con máquinas de estado finito, como por ejemplo la lavadora.
3. Cargas de consumo variable.
4. Cargas de consumo permanente.

Por otro lado, también hay que distinguir la naturaleza de las cargas:

1. Carga resistiva pura: La corriente y la tensión se encuentra en fase. Sólo presenta potencia activa.
2. Carga inductiva: Presenta valores de potencia reactiva positiva, ya que la corriente se encuentra retardada respecto de la tensión ( $-90^\circ$  si es inductiva pura).
3. Carga capacitiva: Presenta valores de potencia reactiva negativa, ya que la corriente se encuentra adelantada respecto de la tensión ( $90^\circ$  si es capacitiva pura).

Así, las señales de las cargas se pueden discriminar en el plano P-Q. En este sentido, la mayoría de los equipos electrónicos son inductivos y las cargas incandescentes suelen ser resistivas.

A su vez, también se pueden encontrar cargas cuya composición se debe a la contribución de dos aplicaciones que están funcionando a la vez. Éstas se denominan cargas compuestas y se pueden analizar como si fueran dos cargas distintas.

Otro método empleado en la extracción de la señal de carga y el reconocimiento de ésta es agrupar las distintas señales por sus características de ondas de corriente y tensión. En este sentido, la onda de la corriente es la que alberga más información en cuanto a la carga sin embargo es complementada por la onda de la tensión que, aunque es más constante incluye cierta distorsión producida por la carga. Así se componen la forma de onda de la admitancia instantánea y la forma de onda de la potencia instantánea, que vienen dadas por las siguientes ecuaciones [2]:

$$IAW(t) = \frac{i(t)}{v(t)} \quad IPW = i(t) \times v(t)$$

*Ecuaciones 2 y 3: Admitancia instantánea y potencia instantánea.*

A su vez, estas curvas también dan información acerca de la distorsión producida por la carga a través de su área encerrada y las no linealidades. En la figura 4 se puede observar algunas de estas trayectorias (durante la conexión y desconexión) para 6 cargas diferentes.

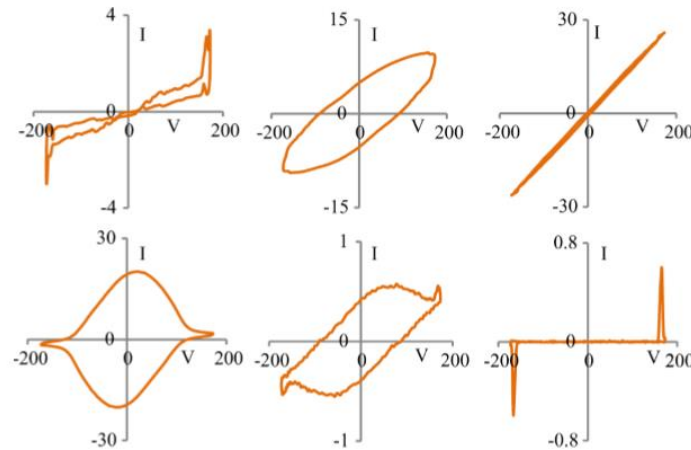


Figura 4: Curvas v-i para 6 cargas diferentes [12]

Muchas de las cargas producen distorsión en la corriente, por lo que es conveniente extraer tanto los armónicos de la misma, ya que éstos contienen gran información acerca de la señal, como la potencia de distorsión. También el 3<sup>er</sup> armónico alberga información acerca de la potencia de consumo, luego ayuda a discernir entre cargas más capacitivas o inductivas y cargas resistivas. Normalmente el análisis de los armónicos se realiza en el dominio de la frecuencia mediante la Transformada de Fourier, sin embargo, también se puede extraer la potencia de distorsión generada por la carga y analizarla.

Por último, otra forma de detectar las distintas señales producidas por las cargas es a partir de las interferencias electromagnéticas creadas, sobre todo por las aplicaciones más modernas. El inconveniente de este análisis es que el ruido producido por este tipo de interferencias ocurre en el rango de los 20 kHz a los 30 MHz luego es muy difícil captarlas ya que el periodo de muestreo debe de ser muy alto [2].

### 5.4.3 IDENTIFICACIÓN DE LA CARGA

Una vez que se ha extraído la señal de las diferentes cargas, hay que identificar qué cargas son las que se encuentran consumiendo potencia en un momento determinado.

Para identificar la carga, se emplean técnicas de *machine learning*, distinguiendo dos procesos: el aprendizaje de la carga y la clasificación.

El aprendizaje puede ser, dependiendo de la aplicación, supervisado o no supervisado. En cuanto al proceso de clasificación de las cargas también depende del tipo de carga y de la aplicación. A continuación, se explican algunas de las técnicas empleadas:

1. Clustering: Es una técnica muy habitual empleada en la clasificación de los estados permanentes de las potencias activa y reactiva. Normalmente, se pueden saber desde un principio el número de clusters que se van a utilizar ya que cada tipo de carga se agrupa bajo un cluster, sin embargo, no siempre es posible.
2. Redes neuronales: Son empleadas, sobre todo, para clasificar los armónicos de la corriente. En este caso también se pueden emplear otros métodos como las máquinas de soporte de vectores (SVM), tomando para el aprendizaje datos agrupados en dos categorías con el objetivo de diseñar un nuevo modelo que clasifique cada carga en una de esas dos categorías.
3. Algoritmos genéticos: Aparte de las redes neuronales, para clasificar las cargas compuestas se emplean algoritmos genéticos. Cuando se produce un evento se extraen todos los datos relacionados con la forma de onda de corriente, los armónicos y las potencias activa y reactiva, con estos datos cada algoritmo elige un tipo de dispositivo. A continuación, los algoritmos cuya elección se aproxime más al tipo de carga se mezclan hasta conseguir el tipo de carga exacta.

Por último, también se puede emplear el ruido generado por las interferencias electromagnéticas para clasificar las cargas. Las cargas basadas en los eventos de encendido producen una interferencia que dura como mucho, microsegundos. Estas pueden ser clasificadas por algoritmos basados en SVM. A su vez, el ruido continuo producido por una interferencia electromagnética es único para cada carga y depende del oscilador interno de la aplicación [2].

## 5.5 SISTEMAS COMERCIALES: CONTADORES Y MEDIDORES EN EL MERCADO.

La mayoría de los contadores actuales trabajan obteniendo una muestra cada 15 minutos, luego la desagregación de energía que se puede conseguir en tiempo real es muy imprecisa [2]. De esta forma, es común que los *Smart meters* que encontramos en el mercado sean capaces de mostrar al consumidor el dato de la potencia consumida en tiempo real, pero que el algoritmo de desagregación de energía se lance por las noches; mostrando así los datos del día anterior.

Éstos, normalmente, están enfocados a que el consumidor final pueda gestionar el uso de la energía eléctrica de forma eficiente ya que; gracias a la desagregación de energía dispone tanto de los datos de consumo de cada electrodoméstico y/o de cada aparato electrónico conectado a la red, así como las horas puntas de consumo.

Por otro lado, algunas aplicaciones también ofrecen al usuario consejos relacionados con el tipo de factura que deben de contratar con la compañía de suministro en función de sus hábitos de consumo.

En cuanto a las compañías de suministro, cada vez son más las que instalan contadores inteligentes. Así en España, se decretó que a partir del 31 de diciembre del 2018 todas las instalaciones con una potencia contratada menor o igual a 15 KW tuvieran un contador inteligente instalado [5]. Estos son bidireccionales, por lo que se pueden enviar la energía al cliente, pero éste puede producir su propia energía y verterla a la red, como por ejemplo en el caso de las instalaciones solares. También, las compañías pueden monitorizar la red y actuar a distancia sobre ella, adelantándose a posibles incidencias.

En España, los contadores en las *Smart Grids* funcionan bajo el protocolo PRIME (*Powerline Intelligent Metering Evolution*). Es un protocolo no propietario, abierto y público que soporta tanto el contador inteligente como la conexión de aparatos a la red eléctrica y las medidas de energía, agua y gas de toda la red del interior de los hogares [7]. Este protocolo está regido por la Alianza PRIME y su medio de transmisión es la PLC.

A continuación, se detallan algunos de los *Smart meters* o dispositivos de terceras partes que se pueden encontrar actualmente en el mercado como las aplicaciones que ofrecen las compañías distribuidoras.

### **5.5.1 PROYECTO STAR DE IBERDROLA DISTRIBUCIÓN**

Iberdrola Distribución bajo el proyecto STAR (Sistema de telegestión y automatización industrial) ha instalado más de 10,8 millones de contadores digitales y ha dotado de capacidad de telegestión a 90.000 centros de transformación [6].

Con este proyecto han conseguido tener monitorizada toda la red de distribución y obtener datos de la red de media tensión. Este hecho les permite que puedan realizar una gestión óptima de toda la red y en caso de avería tener información instantánea del punto donde se ha producido la misma, disminuyendo, por tanto, el tiempo de resolución.

Para gestionar la red inteligente, Iberdrola Distribución tiene actualmente en España 6 centros de control, cada uno de ellos cuenta con 3 sistemas que les permite gestionar la red inteligente de tensión:

1. Sistema SCADA: Controla la red de alta y media tensión.
2. Sistema SPECTRUM: Integra la red y el telemando registrando las incidencias de forma gráfica. [8]
3. Sistema de telegestión: Este es el más nuevo. Se centra en la red de baja tensión y recoge la información de los contadores inteligentes, la cual se encuentra encriptada mediante algoritmos de ciberseguridad.



Por otro lado, todos los nuevos contadores llevan incorporadas las siguientes funcionalidades:

1. Lectura de datos de consumo de forma remota.
2. Se pueden realizar cambios en la tarifa.
3. Se puede conectar y desconectar de forma remota el interruptor de control de potencia (ICP), el cual está integrado en el interior del contador.
4. Obtener datos acerca de la calidad del suministro.
5. Detectar fraudes.

Todas estas funcionalidades permiten al consumidor final tener un ahorro económico en su factura de la luz ya que con los datos digitales obtenidos del contador puede personalizar su factura.

El cliente, también, puede consultar la información del contador a través del área de cliente de la página web de Iberdrola o en las aplicaciones que ha desarrollado la compañía para dispositivos Android y Apple. Así, se puede consultar el consumo casi a tiempo real (el dato de consumo que se ofrece es de un instante antes de que se solicite desde la aplicación), optimizar su factura, gestionar sus expedientes de acometida, recibir información sobre averías en la red entre otras, o revisar sus hábitos de consumo gracias al mantenimiento predictivo que se realiza con los datos obtenidos desde la *Smart Grid*.

### **5.5.2 MIRUBEE**

Esta empresa tiene actualmente dos sistemas de medida en el mercado: “Mirubox mono” y “Mirubox tri”. También disponen de pinzas amperimétricas pequeñas de una corriente nominal de 60 A, pinzas amperimétricas grandes de una corriente nominal de 100 A y conectores magnéticos. Todos estos complementos son compatibles con ambos tipos de medidores.

La principal diferencia que hay entre el Mirubox mono y el Mirubox tri es que el primero de ellos sólo trae incorporada una pinza amperimétrica y permite la conexión de un máximo de 3 pinzas para 3 circuitos monofásicos. El segundo viene de serie con 3 pinzas para la medición de las 3 fases de una instalación trifásica con neutro, pudiendo incorporar una pinza más.

Estos medidores están destinados para el uso doméstico y trabajan conectados al cuadro eléctrico, de donde se alimentan directamente. La medición se realiza mediante la pinza amperimétrica y los datos se almacenan en la nube. La conexión a internet es vía WiFi.

La conexión de éstos al cuadro eléctrico es muy sencilla. Tienen dos partes claramente diferenciadas para medir por un lado la tensión y por otro lado la corriente. Así, primero se elige una de las fases de la instalación y en ella se conecta el neutro y el cable por donde circula la tensión a los bornes correspondientes. En este caso, es indiferente qué fase se elija ya que la tensión que hay en cada una de ellas es la misma. Para finalizar se conecta la pinza amperimétrica al interruptor general. En la figura 5 se muestra la conexión del Mirubox mono con una pinza amperimétrica adicional.

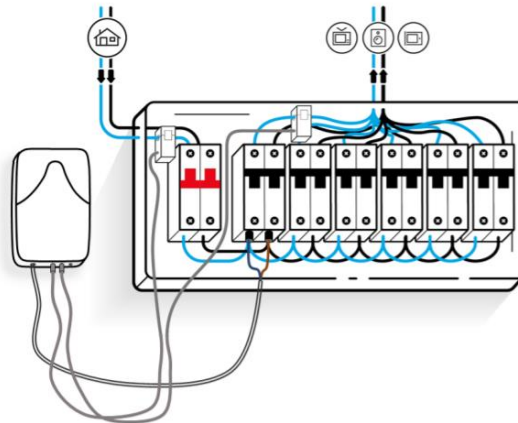


Figura 5 . Instalación final de Mirubox mono con pinza amperimétrica adicional. [9]

Para asegurar que el dispositivo está funcionando correctamente, dispone de dos leds. Uno rojo, que indica que le está llegando la alimentación al dispositivo y su estado normal es parpadeando y otro de ellos azul que se encuentra encendido de forma fija si el dispositivo se encuentra conectado a internet.

Con los datos obtenidos, el dispositivo es capaz de identificar el consumo individual de algunos electrodomésticos y/o aparatos electrónicos empleando un único sensor general. El dispositivo lee y envía datos de las potencias activa, reactiva y tensión cada segundo durante todo el día, resultando una gráfica de 86.000 puntos [10]. Estos datos se almacenan en la plataforma (en la nube) y son los que usará el algoritmo para realizar la desagregación de energía.

Este algoritmo se basa en el reconocimiento de patrones y en la detección de eventos. Para ello, analiza la gráfica obtenida con los datos que ha enviado el medidor y busca saltos en la potencia consumida y las características particulares de cada carga conectada a la red. Éste es lanzado por la noche, por lo que lo que se muestra al usuario son los datos de la desagregación de energía relacionada con el consumo del día anterior.

El algoritmo NILM es capaz de detectar automáticamente las cargas del frigorífico, lavadora, secadora, microondas y la del termo de agua caliente. Para el resto de las cargas hay que entrenar al algoritmo bien de forma asistida o de forma manual.

El entrenamiento asistido se basa en la confirmación de varias preguntas que el sistema lanza al usuario mediante la aplicación o la página web cuando analiza la gráfica del consumo total y sospecha que algún patrón concuerda con alguna carga conocida.

Algunos de los aparatos que necesitan de este entrenamiento son el horno, la lavadora o el lavavajillas.

Por otro lado, el entrenamiento manual se basa en grabar patrones de ciertas cargas. Para ello se le marca al usuario una rutina de conexión y desconexión del aparato. Así el algoritmo puede saber cuáles son los saltos de consumo de cargas como el secador, la plancha o un calefactor.

El resto de los equipos, como vitrocerámicas de inducción, ordenadores o televisores quedan fuera del alcance del algoritmo.

Para que el usuario pueda recibir sus datos de consumo primero tiene que registrar el medidor. Para ello, hay que crear una cuenta nueva en la página web y registrarlo. Si el medidor ya está registrado con otra instalación primero habrá que desvincularlo para poderlo vincular a la instalación deseada.

La información se muestra al usuario a través de su cuenta bien por el área de clientes de la página web o mediante la aplicación de Mirubee disponible para dispositivos Android o Apple. De esta forma, el usuario puede ver el consumo a tiempo real de la instalación, los electrodomésticos que tiene detectados o la desagregación de energía del día anterior. También puede consultar los datos de los electrodomésticos más usados, un histórico de la potencia consumida por meses, los máximos de potencia o incluso un gráfico con los hábitos de consumo repartidos entre los días de la semana y las horas del día, el cual se muestra en la figura 6. En ésta se puede apreciar cómo los picos de consumo se localizan en rojo a partir de las 20:00 horas entre diario y cómo los fines de semana el pico de consumo se amplía casi todo el día.

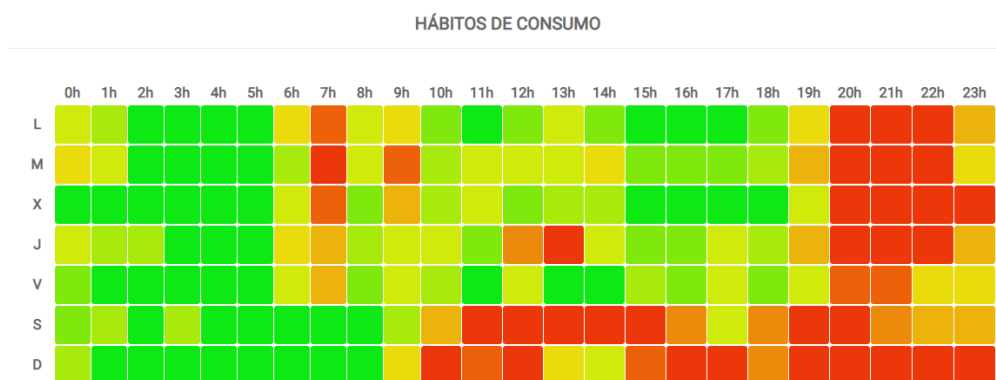


Figura 6: Hábitos de consumo de una vivienda

Por otro lado, la aplicación también ofrece consejos en cuanto a qué tarifa escoger según los hábitos de consumo y muestra el dato de cuánto se habría ahorrado el usuario escogiendo un tipo de tarifa u otra. Desde la aplicación también se puede realizar cambios en la información de la cuenta como por ejemplo registrar un nuevo medidor, modificar el que ya se tiene registrado o modificar la ubicación de la instalación.

A su vez, cuando el mes finaliza se envía un correo electrónico al usuario con un informe que recoge su histórico de consumo para dicho mes.

El medidor Mirubox Mono ha servido en este TFG para estudiar y analizar el funcionamiento y las distintas funcionalidades que ofrece un dispositivo de desagregación de energía (de tercera parte) actual, así como las distintas aplicaciones de las que carece y mejoras que se pueden realizar. En este sentido, en el anexo 1 se recogen los informes de consumo de la instalación en la que se encuentra conectado el medidor desde agosto del 2018 hasta junio del 2019.

Como conclusión este medidor está enfocado a que el usuario pueda hacer un uso eficiente de la energía y gestionar mejor su factura teniendo un mayor control de la energía consumida en su vivienda.

### **5.5.3 INFORMETIS + TEPCO**

Estas dos empresas japonesas ofrecen una suscripción única a un servicio de asistencia en la vivienda basada en la desagregación de energía.

De este modo, han desarrollado una aplicación para móvil a través de la que se realiza la suscripción. Desde la aplicación se pueden consultar la desagregación de energía de los electrodomésticos más usados y el histórico de los datos de consumo a nivel semanal y diario. También dispone de un chat y se pueden enviar mensajes de voz al domicilio, donde se supone que viven las personas que se quieren monitorizar a partir de su uso de las distintas cargas eléctricas.

Por otro lado, también se notifican los cambios a lo largo de días, se manda un correo electrónico con la actividad del día anterior y con la suscripción se incluyen dos visitas a la vivienda suscrita gratuitas al año.

Para poder tomar los datos necesarios para realizar la desagregación se instala en la vivienda un sensor inteligente que mide los datos de consumo de corriente. Estos datos son los que usará el algoritmo y se almacenan en la nube.

La empresa encargada de la instalación del sensor inteligente es Tepco. Ésta también se encarga de la comercialización de la suscripción y del mantenimiento.

El algoritmo encargado de la desagregación de energía está basado en técnicas de Inteligencia Artificial (AI) y ha sido desarrollado por la empresa Informatis. Así, se extrae la señal del consumo total y aplicando dicho algoritmo estiman las formas de onda de cada una de las cargas. Esta empresa también se encarga del desarrollo de la aplicación para móvil.

Esta aplicación es claramente un ejemplo de cómo la desagregación de energía y los sistemas de monitorización de carga no intrusiva pueden servir para la monitorización de personas a través de sus hábitos de consumo.

#### 5.5.4 HOWZ

Howz es un sistema inteligente de monitorización para el hogar. Está compuesto por el kit de monitorización y la suscripción a la aplicación para dispositivos Android o Apple.

El kit de monitorización incluye un enchufe inteligente, un sensor de presencia, un sensor para el movimiento de la puerta de entrada y un *home hub*.

El usuario instala los sensores a lo largo de la vivienda y éstos comienzan a tomar los datos tanto de consumo como de movimiento. Estos datos se envían a un *hub* el cual se encuentra conectado a un *router*. El algoritmo analiza estos datos y aprende las rutinas de comportamiento del usuario en aproximadamente 10 días. Una vez que el aprendizaje ha concluido se envían los datos a la aplicación y si el algoritmo detecta un cambio en la rutina de comportamiento del usuario se envía una notificación por medio de ésta.

Este sistema también está enfocado a la monitorización de personas, sobre todo mayores que quieran seguir manteniendo su independencia. De hecho, la información sobre sus hábitos pueden compartirla con la persona que ellos deseen y que pueda actuar en caso de algún cambio significativo. Sin embargo, este es un método intrusivo ya que hay que desplegar cierta cantidad de sensores por la casa y tampoco hay una monitorización basada en la desagregación de energía en tiempo real.

#### 5.5.5 ENERGENIE: MI|HOME ECO PACK

El Mi|home Eco Pack de Energenie es un kit que está compuesto por un Smart meter con pinza amperimétrica, dos enchufes inteligentes y un *home hub*.

El *hub* da internet a los dos enchufes y al medidor. Así, estos pueden subir a la interfaz los datos que registran. En este sentido, los dos enchufes envían datos de las potencias activa y reactiva, de tensión y frecuencia a la interfaz cada 10 segundos. El medidor se conecta directamente a la instalación eléctrica mediante la pinza amperimétrica que le sirve para recoger los datos de corriente.

Al kit inicial se le pueden añadir interruptores inteligentes, sensores de presencia, adaptadores o válvulas para los radiadores entre otros, con el objetivo principal de que el usuario pueda tener un control mucho mayor en el consumo de su energía.

De esta forma, se ofrece una aplicación para móvil mediante la cual el usuario puede consultar el consumo de potencia de la instalación en tiempo real. Si se incorporan los complementos al kit inicial se le permite al usuario que, por ejemplo, se encienda la luz cuando los sensores de presencia detecten movimiento, o que se encienda la calefacción a una hora determinada. También ofrecen que el cliente pueda crear su propia red de sensores y su propia herramienta a partir de una Raspberry Pi.

En este caso, no se realiza una desagregación de energía ya que lo que se propone no es analizar los hábitos de consumo del usuario para un objetivo dado, sino que se pretende que el usuario pueda tener un control óptimo de su vivienda y la distribución de la energía eléctrica en la misma.

### 5.5.6 PEACEFAIR

Peacefair tiene un módulo cuyo principal objetivo es realizar medidas de tensión corriente y de potencia activa, refrescando los datos cada segundo. Estos datos tienen que ser leídos a través de una interfaz TTL.

Este módulo viene de serie con una pinza amperimétrica que mediría la corriente que está consumiendo un aparato determinado y un software para ordenador que permite leer los datos que ha obtenido el módulo.

El módulo aparte de realizar las mediciones que se han mencionado anteriormente también puede almacenar los datos que ha leído cuando se queda sin alimentación.

Los rangos de funcionamiento que presenta son los siguientes:

1. Tensión: De 80 V a 260 V.
2. Corriente: De 0 A a 100 A.
3. Potencia: De 0 kW a 22 Kw.
4. Energía: De 0 kWh a 9999 kWh.

Este módulo puede resultar interesante si se es capaz de acceder a los datos obtenidos directamente desde el hardware, leerlos y analizarlos de cara a distintas aplicaciones de desagregación de energía.

### 5.5.7 WIBEEE

Wibee es un analizador de consumo con conexión inalámbrica vía WiFi que muestra los datos instantáneos e históricos del consumo eléctrico mediante su aplicación desarrollada para *smartphones*, *tablets* u ordenadores [11].

Actualmente hay 3 modelos en el mercado: Wibeec monofásico, Wibeec trifásico y Wibeec Max.

El aparato hay que instalarlo directamente en el cuadro eléctrico de la vivienda, de forma que los conectores magnéticos del interruptor del cuadro queden junto al cuerpo del medidor. Tiene dos LEDs que indican si el equipo está conectado y enviando datos, en este caso el led azul se encuentra encendido fijamente y el led rojo parpadeando cada 1s.

Una vez instalado el medidor y cuando su corriente sea de 10 A hay que calibrarlo. Para ello, simplemente hay que medir la corriente de cada una de las líneas con una pinza amperimétrica e introducirla en la aplicación.

A su vez, Wibeee mide los valores de las potencias activa, reactiva y aparente, el factor de potencia, la frecuencia, los valores de tensión y corriente consumidos y las emisiones de CO<sub>2</sub>.

El equipo tiene dos modos de funcionamiento:

1. Modo aplicación: El equipo recopila los datos y los almacena en la nube. Estos datos se muestran tanto en la aplicación para móvil y en la plataforma web como en el servidor interno del que dispone éste. En la aplicación se pueden visualizar y configurar las medidas comentadas anteriormente. En la plataforma web el usuario puede consultar los datos de consumo del mes actual y del mes anterior y también se muestra una previsión de consumo con un objetivo. Otra de las utilidades que tiene la plataforma es que muestra un histórico de consumo en un periodo definido por el usuario y las últimas medidas enviadas por el equipo. Por último, en la plataforma también hay una sección de ayuda para intentar solventar las dudas que le puedan surgir al usuario.
2. Modo compartido directo: En este modo los datos sólo son accesibles mediante el servidor web interno. Una vez que se accede al servidor éste muestra los datos instantáneos recogidos por el equipo. También permite configurar los parámetros de red y de IP y acepta peticiones HTTP GET para realizar las consultas de la configuración del Wibeee.

En este caso tampoco se realiza una desagregación de energía, simplemente se le muestra al usuario unos datos de consumo y de objetivos de cara a que el usuario pueda gestionar de forma óptima la energía.

## 5.6 HARDWARE

El diseño hardware de un E-meter consta de tres grandes bloques fundamentales, los cuales se pueden apreciar en la figura 7:

1. La MCU: Es la unidad básica de procesamiento de datos, el microcontrolador el cual incorpora ADCs para la conversión de los datos obtenidos, e incluso DACs.
2. El sistema de alimentación: Algunos son sensores no intrusivos, alimentándose así de la propia fuente de la que obtienen los datos. También pueden incorporar pilas o baterías como sistema de alimentación.
3. Bloque de comunicaciones: En este bloque se incluyen todos los protocolos y tecnologías de comunicación usadas para almacenar los datos y enviarlos al usuario.

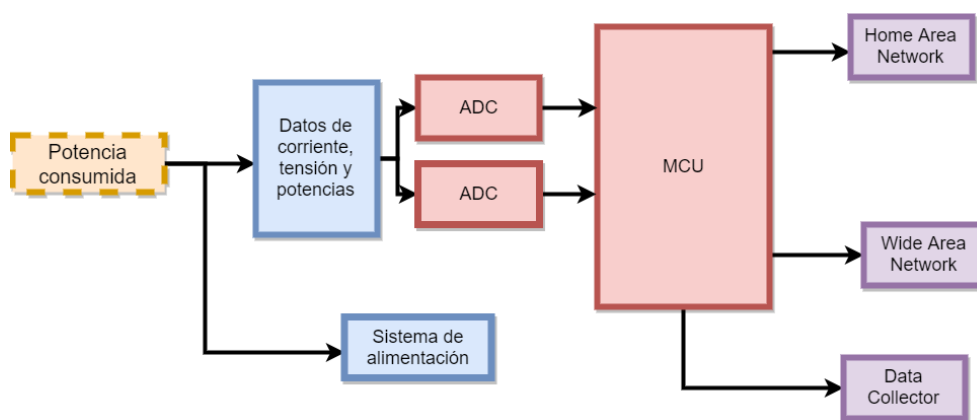


Figura 7: Diagrama de bloques del diseño HW de un E-Meter

La Home Area Network (HAN), es un área local que permite que los distintos aparatos digitales que se pueden tener en un hogar se comuniquen y operen entre sí. Algunas de las tecnologías y protocolos de comunicación habilitados en este tipo de sensores son: Zigbee, WiFi o 6LoWPAN.

A su vez, las distintas HAN se interconectan a la Wide Area Network (WAN) mediante un router. Así, por ejemplo, todos los datos se pueden almacenar en la nube para trabajar con ellos y ofrecerle al usuario los distintos datos de consumo.

El diseño hardware del presente TFG se basa en el chip ADE9153A de Analog Device. Éste se encuentra integrado en la tarjeta “EV-ADE9153ASHIELDZ”, la cual se alimenta directamente de la energía que llega a la instalación.

Para el procesado de los datos se emplea un módulo de Arduino Uno ya que esta tarjeta dispone de una configuración de pines compatible con placas de Arduino Uno y Zero. El módulo de Arduino se encuentra alimentado a 5 V que recibe directamente de la tarjeta de evaluación.

En los siguientes apartados se explican de manera detallada tanto la tarjeta de adquisición como el diseño hardware propuesto.



### 5.6.1 TARJETA DE ADQUISICIÓN “EV-ADE9153ASHIELDZ”.

Se ha escogido esta tarjeta ya que permite trabajar directamente con los datos instantáneos de corriente y tensión y poder procesarlos para poder realizar los cálculos para la detección de eventos, empleando un microcontrolador que se comunica con ésta a partir del SPI. A su vez, permite muestrear los datos a una frecuencia de 4 kHz; frecuencia suficiente como para poder analizar el transitorio de las señales producidas por las distintas cargas, que es realmente donde reside la mayor información.

Como se ha especificado en apartados anteriores esta tarjeta lleva integrado el chip ADE9153A, que permite medidas de energía monofásicas o trifásicas (aunque esta tarjeta de evaluación usada solo dispone de sensores para la versión monofásica). Otras de las funcionalidades de este chip es que se autocalibra obteniendo medidas con una tolerancia del 1% sobre el rango dinámico.

En la figura 8 se aprecia la tarjeta, señalando las partes más importantes de su diseño:

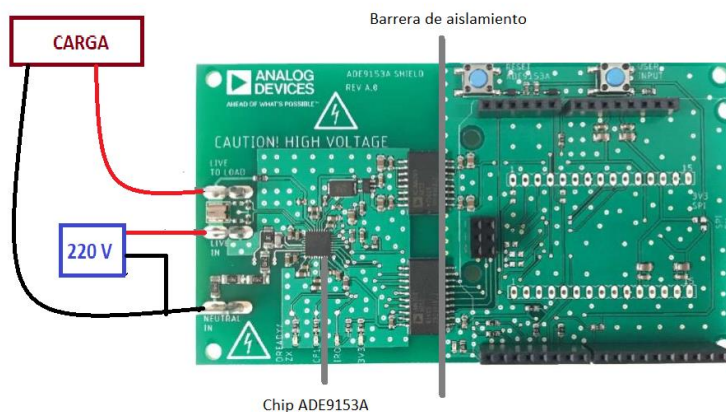


Figura 8: Tarjeta de adquisición EV-ADE9153ASHIELDZ

Según se puede observar en la figura 8, la tarjeta tiene dos partes claramente diferenciadas y separadas por una barrera de aislamiento, que permite que se puedan trabajar con los datos de forma segura.

En dicha figura, también se refleja la conexión de la carga a la tarjeta. La carga va conectada a la primera parte de la tarjeta la cual se alimenta de la propia tensión de la red.

La tarjeta puede extraer los datos de la corriente y de la tensión a través de los sensores que dispone a la entrada y del neutro, tal y como se puede observar en la figura 8.

El sensor de corriente se compone de una resistencia shunt de 1 mΩ. El valor nominal de la corriente es de 5 A rms, y el valor máximo de 10 A rms. También dispone de un amplificador de ganancia programable cuya ganancia es de 16. [15]

El sensor de tensión está compuesto por varios divisores de tensión con un factor de atenuación de 1001. La tensión nominal máxima soportada es de 240 V rms. [15]

En la tabla 1 se describen la relación de pines entre los módulos de Arduino UNO y Zero y el chip ADE9153A.

ADE9153A	ARDUINO	TIPO
SCLK	ICSP-SCK	SPI
MISO	ICSP MISO	SPI
MOSI	ICSP MOSI	SPI
SS	GPIO 8	SALIDA SPI
RESET	GPIO 4	SALIDA
USER BUTTON INPUT	GPIO 5	ENTRADA
ZX/DREADY/CF2	GPIO 2	INTERRUPCIÓN
IRQ	GPIO 3	INTERRUPCIÓN
NO APLICABLE	ALIMENTACIÓN 5 V	ALIMENTACIÓN
NO APLICABLE	ALIMENTACIÓN 3,3 V	ALIMENTACIÓN

Tabla 1: Relación de pines entre ADE9153A y Arduino

### 5.6.2 CHIP AD9153A

En este apartado se explica con más profundidad las distintas funcionalidades y características del chip en el que se basa la tarjeta, los diferentes periféricos que lo integran y el bloque de metrología.

En la figura 9 se muestra un diagrama con los bloques funcionales que integran la tarjeta y que se explican a continuación.

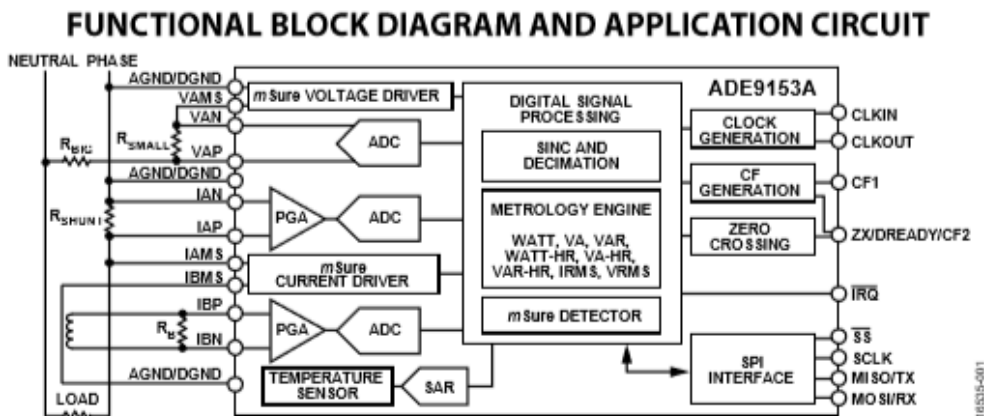


Figura 9: Diagrama de bloques ADE9153A [13]

### 5.6.2.1 Autocalibración: mSure

Una de las funcionalidades principales que ofrece este chip es la función mSure. Esta función permite que se realicen la calibración de la corriente y de la tensión de forma automática, sin necesidad de emplear un patrón.

El proceso de autocalibración se basa en estimar la función de transferencia de los sensores de corriente y tensión y posteriormente ajustar la ganancia de dicha función de transferencia. Caben ser destacados dos conceptos que tienen especial relevancia en este proceso:

1. Velocidad de convergencia: Es el tiempo que el proceso de autocalibración tarda en conseguir cierto nivel de exactitud. Así, la velocidad incrementa de forma logarítmica conforme al grado de exactitud requerido.
2. Precisión absoluta: Tiene en cuenta la precisión de la referencia del proceso de autocalibración. De esta forma, a mayor tiempo de ejecución de la función mSure, mayor será la precisión.

Una vez que el medidor se encuentre conectado, el proceso se puede ejecutar de forma secuencial en cada canal, pudiendo configurar un tiempo determinado para la ejecución en cada canal.

También, los canales de corriente se pueden configurar en dos modos de consumo de potencia: Modo normal o modo turbo, en el cual el consumo de corriente es el doble y la velocidad de convergencia 4 veces mayor respecto al modo normal.

Aparte de la versión de la tarjeta empleada, existe la versión ADE195B. Ésta permite al usuario monitorizar la exactitud del proceso de calibración e incorpora un monitor de potencia inversa.

### 5.6.2.2 Conversor analógico-digital: ADC

En la figura 10 se puede apreciar un diagrama en el que se muestran los distintos bloques que integran la conversión analógica-digital en rasgos generales en el chip ADE9153.

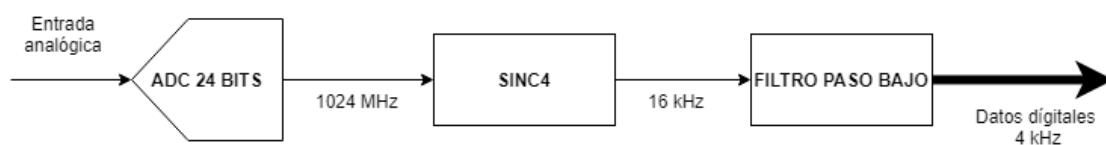


Figura 10: Diagrama de bloques de la conversión A-D

El ADE9153A incorpora tres ADCs, dos de ellos convierten los datos adquiridos de la corriente y un tercero que convierte los datos de la tensión. Estos ADCs son delta-sigma de segundo orden de 24 bits.

El canal de corriente A admite una entrada diferencial con la señal IAP, así la tensión en modo común entre las señales de los pines IAP e IAN no deben superar los  $\pm 0,5$  V y las señales de entrada en ambos pines alcanzaran como máximo 0,1125 V.

El canal de corriente B incorpora entradas totalmente diferenciales. Las señales de los pines IBP e IBN deben estar comprendidas entre los 0,35 V y los 1,45 V.

Para ambos canales el rango de entrada del ADC es de  $\pm 1$  V, sin embargo, éstos incorporan amplificadores con ganancia programable que permiten atenuar la tensión del ADC. El canal A permite unos valores de ganancia de 16, 24, 32 o 38,4 y el canal B de 1, 2 o 4.

Por otro lado, el canal de tensión es una entrada pseudo-diferencial. La tensión en modo común es de 0,8 V, permitiendo un valor máximo de entrada de 1,3 V. Esta tensión en modo común se produce cortocircuitando los pines VAMS y VAN y es necesaria para la función de autocalibración de la entrada de tensión.

Para los canales de corriente como el de tensión se ha escogido un filtro *antialiasing* externo de 10 kHz y atenuar las señales fuera de banda y evitar errores de fase entre la corriente y la tensión. También se disponen de filtros *antialiasing* internos que sirven para filtrar las frecuencias procedentes de los campos de radiofrecuencias, ya que según la normativa IEC 61000-4-3 las aplicaciones de medición de energía requieren que el medidor sea inmune a campos de radiofrecuencia externa 30V/m desde los 80 MHz hasta los 10 GHz [13].

Respecto a la función mSure, para que ésta funcione correctamente tanto en los canales de corriente y de tensión es necesario que se cumplan los siguientes puntos: [13].

1. La resistencia de los cables y derivaciones tiene que ser menor a  $2,2 \Omega$  en los canales A y B.
2. La resistencia total del divisor de tensión tiene que ser mayor de  $900 \text{ k}\Omega$ .
3. Para que la autocalibración sea precisa, la tensión de la línea debe ser menor a 440 Vrms.

Por último, dentro de las características de trabajo de este chip cabe resaltar que admite una resistencia interna de 1,25 V y tiene un coeficiente de temperatura de valor típico  $\text{CCT} = \pm 5 \text{ ppm}/^\circ\text{C}$  y de valor máximo  $\text{CCT}_{\text{max}} = \pm 30 \text{ ppm}/^\circ\text{C}$ . También se puede conectar una referencia externa entre los pines REFIN y AGND. [13]

### 5.6.2.3 Serial Peripheral Interface: SPI

La comunicación entre el ADE9153A y el Arduino se hará mediante el SPI que incorpora el chip. En el ADE9153A el SPI siempre será el esclavo, siendo compatible con operaciones de lectura y escritura de 16 bits o de 32 bits. La máxima frecuencia de reloj a la que puede funcionar el SPI es de 10 MHz.

Hay tres ciclos en el protocolo de comunicación del SPI, teniendo en cuenta que los datos entran y salen del ADE9153A en los flancos de bajo nivel de la señal de reloj y son mostrados por el máster en los flancos de nivel alto. Las tres etapas son las siguientes:

1. Cuando el máster inicia la comunicación se envía un comando de 16 bits (comando de cabecera) en el que se especifica si la operación es de escritura o lectura. Este comando también tiene reservados los 12 bits de mayor peso para indicar la dirección del registro que va a ser leído o escrito.
2. En el siguiente ciclo se envía el dato de 16 bits o de 32 que va a ser escrito en el registro o leído desde él. Para que se inicie cualquiera de las dos operaciones y la transmisión del dato sea correcta el *chip select* del ADE9153A (SS) tiene que estar a nivel bajo durante todo el proceso.
3. Para la operación de lectura se envía un *check* de confirmación (CRC) de que el dato ha sido leído. Cuando una operación sea completada el máster debe cambiar el SS a nivel alto para liberar el bus del SPI.

En la figura 11 se puede observar dos ejemplos del funcionamiento en modo escritura y en modo lectura del SPI.

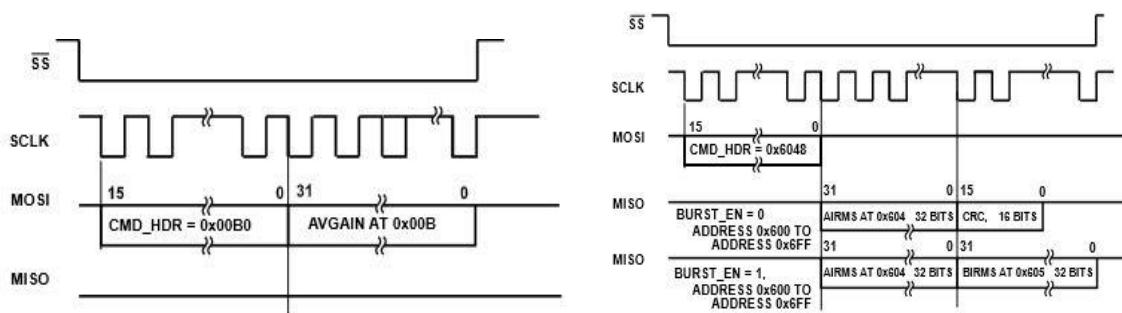


Figura 11: Ejemplo de las operaciones de escritura y lectura en el SPI [13]

La operación de escritura tendrá lugar cuando el bit 3 del comando de cabecera se encuentre a nivel bajo y la operación de lectura cuando éste se encuentre a nivel alto.

Por otro lado, el SPI del ADE9153A incorpora a parte del modo convencional de lectura, el modo *burst*. El funcionamiento de este modo es el mismo que el de la lectura normal, sin embargo, si éste se activa a nivel alto mediante software permite que se puedan leer varios registros después de que se envíe el comando de cabecera. El proceso de lectura acabará cuando el máster cambie a nivel alto el SS. Este modo está accesible en los registros que se encuentran desde la dirección 0x600 hasta la 0x6FF.

Por último, cabe resaltar que el SPI incluye tres registros de verificación: El LAST\_CDM, que indica que se ha recibido el comando de cabecera y los LAST\_DATA de 16 y 32 bits, los cuales se actualizan cada vez que una operación de lectura se haya completado, ya sea en modo *burst* o en una lectura normal, o cada vez que se reciban los datos de escritura.

#### 5.6.2.4 Universal Asynchronous Receiver-Transmitter: UART

El ADE9153A incorpora también una UART con dos pines para la transmisión y la recepción de los datos. Es compatible con datos de 16 bits o de 32 y para operaciones de lectura y escritura. Funciona con un *baud rate* de 4800 baudios, aunque también se puede seleccionar un valor de 115200 baudios para alcanzar mayor velocidad en la transmisión y en la recepción de los datos.

La comunicación se basa en un *frame* de 11 bits, con un bit de comienzo, 8 bits reservados para los datos, un bit de paridad y otro de stop [13].

Cuando se inicia la comunicación se envían dos comandos de cabecera que contienen la dirección del ADE9153A que puede ser accesible, otro bit que indica la operación (0 para escritura, 1 para lectura) y otro para el *checksum*. Posteriormente se envían los datos de transmisión o de recepción y otro *checksum* que es opcional.[13]

#### 5.6.2.5 Interrupciones

El ADE9153A tiene un registro de máscara de 32 bits para las interrupciones, así como un registro de estado. La máscara sirve para poder habilitar o deshabilitar las interrupciones. Para ello se tiene que establecer el bit deseado a 1, habilitando así la interrupción y borrar éste para deshabilitarla. Por otro lado, el registro de estado indica si un evento que ha ocurrido puede generar interrupción o no.

A su vez, se disponen de dos pines que pueden ser usados como interrupciones: IRQ activo a nivel bajo y el ZX/DREADY/CF2, que indica el estado de las interrupciones habilitadas ya que en función de las señales internas cambia a nivel alto o a nivel bajo [13].

Las interrupciones se encuentran divididas en dos niveles: Nivel 1 en el que se encuentra el registro de estado y nivel 2 en las que se encuentran los siguientes registros: CHIP\_ESTATUS, EVENT\_STATUS, MS\_STATUS\_IRQ [13].

Hay interrupciones que se dan, aunque su bit correspondiente en el registro de máscara es 0, es decir no son enmascarables. También hay otras que se usan en conjunto con otros registros de estado.

Por último, cabe destacar que el chip también ofrece un conjunto de interrupciones cuya función es alertar al usuario de algún tipo de fallo durante el proceso de autocalibración.

### 5.6.2.6 Medidas

El ADE9351 puede captar los valores instantáneos, eficaces y de pico de la corriente consumida y la diferencia de fase entre la tensión y la corriente, a una frecuencia de muestreo de 4 KHz. Tanto los valores instantáneos como los valores eficaces son recogidos en dos canales: A y B.

Para obtener los valores instantáneos y las formas de onda de la corriente calcula la ganancia del canal obtenido y se aplica la compensación de fase y de ganancia. Con el valor de la tensión, se calculan la potencia activa total, la reactiva y la de distorsión y también el valor eficaz de la corriente.

A continuación, en la tabla 2 se recogen los registros relacionados con el cálculo de la corriente y su descripción.

REGISTRO	DESCRIPCIÓN
AI_WAV	Corriente instantánea en el canal A
BI_WAV	Corriente instantánea en el canal B
AIRMS	Valor eficaz de la corriente del canal A
BIRMS	Valor eficaz de la corriente del canal b
IPEAK	Valor de pico de la corriente del canal muestreado
ANGL_X_X	Margen de fase entre la corriente y la tensión.

Tabla 2: Registros relacionados con la corriente en el ADE9153A

Los datos de tensión también son captados a la misma frecuencia de muestreo, por un solo canal. En la tabla 3 se muestran los registros relacionados con los valores de tensión y su descripción

REGISTRO	DESCRIPCIÓN
AV_WAV	Valor instantáneo de la tensión
AVRMS	Valor eficaz de la tensión
VPEAK	Valor de pico de la tensión.
APERIOD	Medida del periodo de la línea.
ANGL_Ax_xL	Ángulo de fase que hay entre la corriente y la tensión o viceversa

Tabla 3: Registros relacionados con la tensión en el ADE9153A

En cuanto la potencia activa, ésta es calculada para cada fase multiplicando las formas de onda de los valores instantáneos de corriente y tensión y aplicando al resultado un filtro paso bajo. Dicho filtro extrae la potencia activa total atenuando los armónicos de 50 Hz a 64 dB.

Para calcular la potencia reactiva también se multiplican las formas de onda de los valores instantáneos de la tensión y la corriente (ésta desplazada 90°) y el resultado se filtra mediante un filtro paso bajo. Posteriormente se aplica el valor del registro APGAIN para realizar la corrección de ganancia. Para que el cálculo de este valor sea preciso es indispensable inicializar de forma correcto el valor de la frecuencia de trabajo y el valor nominal medido en el canal de tensión.

La potencia aparente se calcula multiplicando los valores eficaces de la corriente y la tensión y aplicando la corrección de ganancia pertinente.

Las medidas de energía son actualizadas cada 4 kHz y se pueden almacenar hasta 106 segundos de valores acumulados. En la tabla 4 se muestran los registros relacionados con las medidas de energía y potencia.

REGISTRO	DESCRIPCIÓN
AWATT	Filtro paso bajo para filtrar la potencia activa total.
AWATT_ACC	Potencia activa total acumulada.
AWATTHR	Energía activa total acumulada.
APERIOD	Medida del periodo de la línea.
APF	Factor de potencia.

Tabla 4: Registros relacionados con la energía y la potencia en el ADE9153A

A su vez, en la figura 12 se puede ver cómo los valores de los registros AI\_WAV y AV\_WAV se emplean para calcular los valores eficaces, las potencias y como estos resultados se acumulan en los registros relacionados con las medidas de energía. [13].

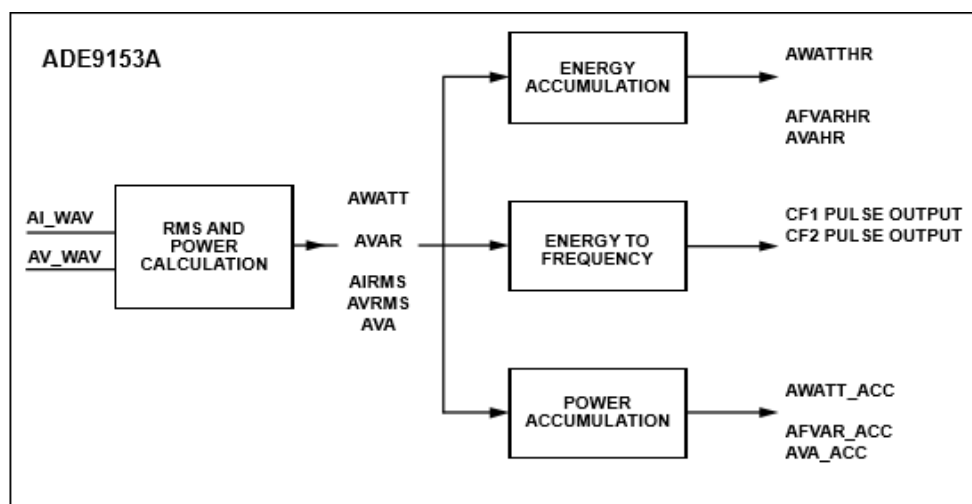


Figura 12: Cálculos de energía y potencia a partir de los valores instantáneos [13]



En el ADE9153A hay ciertas fuentes de error. Así, el sensor de corriente introduce un error de ganancia, se dan pequeñas variaciones en la referencia de tensión y también está presente un error de fase.

A su vez, para configurar el valor del registro de ganancia se aplica la siguiente expresión:

$$xIGAIN = (Ganancia\ de\ corriente - 1)2^{17}$$

*Ecuación 4: Valor registro de ganancia.*

Se puede usar el registro de ganancia de corriente para cambiar el signo de la propia magnitud de corriente. Para ello, hay que cambiar el valor del registro de ganancia sustituyendo la ganancia de corriente por -1 en la ecuación 4. Esto puede ser útil si se ha conectado el sensor de corriente al revés [13].

En cuanto a la compensación de fase, se utiliza un filtro digital que ajusta la fase con una tolerancia de  $\pm 0,001^\circ$ . A su vez, los canales A y B de corriente tienen incorporados un registro de compensación de fase, APHASECAL; para corregir el error de fase. De esta forma la corriente se retrasa de acuerdo con el valor configurado en este registro.

Por otro lado, el canal de tensión también se retrasa a 4 KHz, 4,5° a 50 Hz, ya que la calibración de la fase se encuentra en el rango de -15° a 4,5° a 50 Hz.

Así la corrección de fase entre la corriente de entrada y la tensión es el resultado de combinar ambos retardos.

En cuanto a la tensión, el registro de ganancia de la tensión sirve para calibrar la tensión en cada fase, dicho registro tiene las mismas prestaciones que el registro de ganancia de la corriente.

Por último, cabe destacar el error que se da en la potencia total activa cuando se aplica el filtro paso bajo mencionado anteriormente, obteniendo una variación de  $\pm 0,062\%$ .

## 5.7 SOFTWARE Y DATOS

En este apartado se explica la fase experimental del presente TFG, recogiendo el software que se ha empleado para recabar todos los datos necesarios, así como las modificaciones que se han realizado sobre éste, el algoritmo desarrollado para procesar los datos y detectar los distintos eventos que se dan en la red y por último las distintas pruebas que se han llevado a cabo y los resultados que se han obtenido.

### 5.7.1 SOFTWARE EMPLEADO.

Analog Devices facilita al usuario un software en C++ que corre sobre Arduino y que permite inicialmente, consultar los valores de tensión y corriente eficaces, las potencias activa, reactiva y aparente, el factor de potencia, la frecuencia que se ha configurado y la temperatura del ambiente. Estos valores se muestran a través del monitor serie que incorpora el IDE de Arduino con un periodo de 2 segundos.

Este software se basa de dos librerías, del programa principal y del código fuente. A continuación, se explican brevemente cada uno de ellos:

1. ADE9153A.h: En esta se incluye toda la tabla de vectores con las posiciones en memoria de los distintos registros que componen el chip ADE9153A.
2. ADE9153AAPI.h: Esta librería recoge primero la configuración de los distintos registros y de las distintas constantes, así como se definen los valores usados para la calibración de los valores eficaces. También incluye la definición de las estructuras y las variables globales que son usadas en el código fuente y en el programa principal para el acceso a los distintos registros.
3. ADE9153AAPI.cpp: En el código fuente se realiza la inicialización del SPI, así como se programan las operaciones de lectura y escritura de éste para 16 y 32 bits. También se inicializan los registros y se accede al valor de los registros relacionados con la energía, valores eficaces, potencias y temperatura. A su vez, también se programan todas las funciones relacionadas con la calibración de cada uno de los canales tanto de corriente como el de tensión.
4. ADE9153AAPI\_Test.ino: Este fichero es el programa principal para Arduino. En éste se definen y configuran los pines usados de la tarjeta de Arduino y la velocidad del SPI y la del puerto serie. En el bucle principal se programa la parte de interfaz con el usuario, es decir, se llama a todas las funciones de autocalibración, mostrando por el monitor serie cuando se ha terminado el proceso y se muestran también todos los valores descritos anteriormente cada 2 segundos. A su vez, también se configura el botón de reset y el parpadeo de un led que indica que el programa está corriendo sobre la placa de Arduino.

Para conseguir adquirir los datos de forma óptima y poder procesarlos se ha tenido que modificar alguna parte de cada uno de estos archivos. En los siguientes apartados se explica de forma más concisa cada uno de los ficheros incluyendo las modificaciones que se han realizado sobre el código proporcionado.

#### 5.7.1.1 Librerías

Como se ha especificado anteriormente hay dos librerías, de las cuales ha sido necesario modificar alguna inicialización de la ADE9153AAPI.h. En la librería que contiene las posiciones en memoria de los registros no se ha realizado ningún cambio.

En cuanto a la librería que incluye la inicialización de los registros simplemente se ha modificado el valor inicial del registro que permite seleccionar el valor de la frecuencia de la red. Así, el valor del bit 4 del registro *ACCMODE* permite seleccionar dicha frecuencia, si está a 1 será para una frecuencia de 60 Hz y si está a 0 para 50 Hz.

Para el presente TFG se ha seleccionado la frecuencia de 50 Hz ya que la red eléctrica española trabaja a dicha frecuencia.

El resto de los parámetros de inicialización se han dejado los que venían por defecto aun así es interesante explicar los valores de la ganancia de corriente y los valores empleados para la función de calibración.

En cuanto al valor de ganancia de la corriente se configura en el registro *AI\_PGAGAIN*. El valor por defecto que está programado para una ganancia de 16 siendo el valor recomendado por el fabricante para obtener unos resultados en las medidas óptimos. Sin embargo, este valor se tendría que modificar si se espera que la corriente máxima medida sea mayor de 10 A. En la presente aplicación, no se espera que el conjunto de las cargas conectadas supere dicho valor, además las instalaciones normalmente están diseñadas para aguantar una corriente máxima de 10 A por lo que no es necesario modificar el valor que tiene programado por defecto el registro.

Por otro lado, esta librería también recoge el valor de las constantes relacionadas con la función de calibración. Los valores de estas constantes de calibración son siempre los mismos, y se pueden calcular aplicando las siguientes fórmulas, según la magnitud:

1. Corriente eficaz:

$$CAL\_IRMS\_CC = \frac{I_{max} \times A_{head}}{52725703}$$

*Ecuación 5: Valor para la calibración de la corriente eficaz.*

Siendo  $A_{head}$  el margen que tiene el canal de corriente a máxima escala, tomando el valor de 4,42 A e  $I_{max} = 10$  A

2. Tensión eficaz

$$CAL\_VRMS\_CC = \frac{V_{max} \times V_{head}}{26362852}$$

*Ecuación 6: Valor para la calibración de la tensión eficaz.*

Siendo  $V_{head}$  el margen que tiene el canal de tensión a máxima escala, tomando el valor de 1,47 A y  $V_{max} = 240$  V, que es el valor máximo de tensión que aguanta el chip.

3. Potencias activa, reactiva y aparente:

$$CAL\_POWER\_CC = CAL\_IRMS\_CC \times CAL\_VRMS\_CC \times \left(1 + \frac{APGAIN}{2^{27}}\right)$$

*Ecuación 7: Valor para la calibración de las potencias.*

Por último, en esta librería también se han declarado las distintas estructuras necesarias para poder almacenar y trabajar con los valores de los diferentes registros.

### 5.7.1.2 Código fuente.

En el código fuente vienen programadas todas las funciones de acceso a los registros, las funciones de calibración y las funciones relacionadas con el SPI.

Las funciones de acceso a los registros simplemente se basan en acceder al registro deseado y multiplicar su valor por la constante de calibración que corresponda. Posteriormente el resultado se almacena en una variable para poder trabajar con el mismo.

En cuanto al SPI como se ha especificado al principio del apartado se programan las funciones relacionadas con las operaciones de lectura y escritura tanto de 16 bits como de 32 bits.

Respecto a la calibración se han programado tanto las funciones de inicio y fin de la calibración de cada canal, la lectura de los registros relacionados con la función mSure y la función en la que se aplica los valores obtenidos de la calibración a las ganancias de tensión y corriente.

La calibración se puede realizar una vez que el sistema se encuentre encendido y el bit de estado se encuentre a nivel alto. Así las funciones que se han programado para llevar a cabo el inicio de la calibración de cada canal simplemente consultan el valor de dicho bit durante un tiempo determinado y si éste se encuentra a 1 escriben el valor que corresponda para cada canal en el registro de configuración de la calibración. Estos valores se pueden encontrar en las notas de aplicación del chip.

La calibración no sólo se aplica a los valores que se almacenan en los registros de medidas, sino que también se aplica a los valores de las ganancias de los canales de tensión y corriente. Para aplicar la función mSure a las ganancias, además de emplear las constantes de calibración que se han explicado en el apartado anterior, también se necesitan los valores que se obtienen durante el proceso y que se almacenan en los registros *MS\_ACAL\_AICC* y *MS\_ACAL\_AVCC* para la corriente y la tensión respectivamente. De esta forma, se accede al valor de éstos, se almacena en una variable y posteriormente se accede a los registros que indican si la calibración ha terminado tanto en el canal de corriente como en el canal de tensión.

Una vez que se comprueba que todo el proceso de calibración ha finalizado se aplican los valores obtenidos a las distintas ganancias mencionadas anteriormente, escribiendo el valor resultante en los registros que almacenan el valor de las ganancias de corriente y tensión.

En este fichero se han tenido que modificar distintas partes del código. Primero, se comprobó que cuando se capturaban datos relacionados a una carga resistiva los valores de la potencia activa y del factor de potencia eran negativos, lo que indicaba que había un error en el código bien en la lectura de los registros o bien en el cálculo de los valores de las ganancias de corriente y tensión aplicando los valores resultantes de la calibración.

Así se comprobó que cambiando el sentido de la ganancia de corriente en la función que calcula su valor el error se solventaba. En la figura 13 se muestra esta parte de código aplicando la solución comentada.

```
bool ADE9153AClass::ApplyAcal(float AICC, float AVCC)
{
    int32_t AIGAIN;
    int32_t AVGAIN;

    AIGAIN = -(AICC / (CAL_IRMS_CC*1000)) - 1) * 134217728;
    AVGAIN = (AVCC / (CAL_VRMS_CC*1000) - 1) * 134217728;

    SPI_Write_32(REG_AIGAIN, AIGAIN);
    SPI_Write_32(REG_AVGAIN, AVGAIN);
}
```

Figura 13: Cálculo de las ganancias de tensión y corriente.

Por otro lado, también se ha tenido que cambiar una sentencia dentro de la función que programa el inicio del proceso de calibración del canal de corriente. Como se ha explicado anteriormente, primero hay que leer el valor del bit de estado durante un tiempo y comprobar que el sistema se encuentra preparado para que la comience la calibración. Así la sentencia en la que se lee dicho bit se ha incluido dentro de un bucle while, tal y como se muestra en la siguiente figura.

```
bool ADE9153AClass::StartAcal_AINormal(void)
{
    uint32_t ready = 0;
    int waitTime = 0;

    //ready = SPI_Read_32(REG_MS_STATUS_CURRENT); //Read system ready bit

    while((ready&0x00000001)==0)
    {
        ready = SPI_Read_32(REG_MS_STATUS_CURRENT);
        if(waitTime>11)
        {
            return false;
        }
        delay(100);
        waitTime++;
    }

    SPI_Write_32(REG_MS_ACAL_CFG, 0x00000013);
    return true;
}
```

Figura 14: Función programada para empezar la calibración del canal de corriente

La función relacionada con el canal de tensión no ha sido necesaria modificarla ya que estaba programada tal y como se muestra en la figura 14.

También se han inicializado algunas variables locales que no estaban inicializadas.

### 5.7.1.3 Programa principal

En el fichero ADE9153AAPI\_Test.ino se programa el proyecto para Arduino y como se ha explicado anteriormente, básicamente se configuran los pines que se emplean de la tarjeta de Arduino, la velocidad del puerto serie y del SPI y la interfaz con el usuario para mostrar los datos deseados. Así en la figura 15 se puede observar un flujograma que describe el desarrollo seguido para la inicialización de los distintos parámetros.

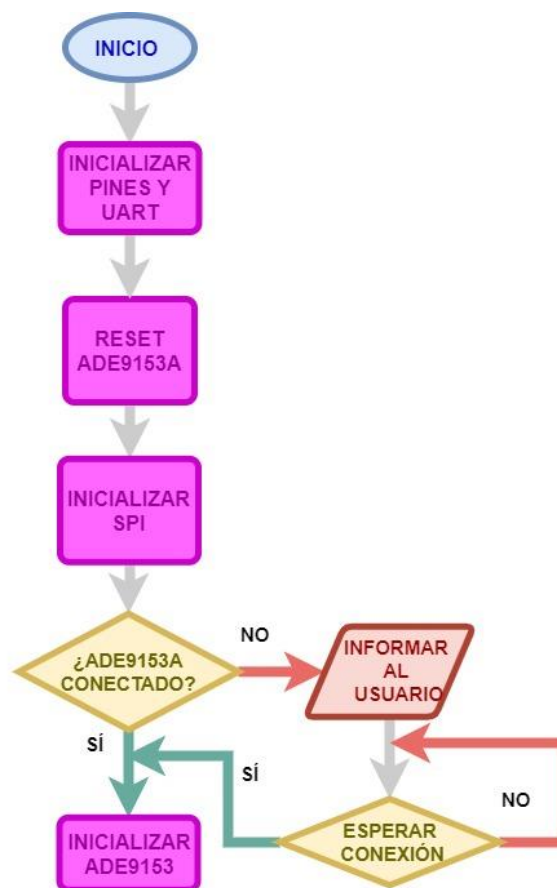


Figura 15: Flujograma programa principal: Inicialización parámetros.

Uno de los pines que se configuran, es el *USER\_INPUT* como entrada. Este es un botón que incorpora la tarjeta que en el momento que el usuario lo presiona comienza todo el proceso de calibración que se ha explicado anteriormente.

En el bucle principal, primeramente, se llama a la función de lectura y escritura. Esta función simplemente accede a los valores que se han almacenado en las estructuras y los almacena en la MCU del Arduino. Una vez almacenados los muestra a través del monitor serie.

También en el bucle principal se realiza la autocalibración, mostrando al usuario un mensaje cada vez que comienza el proceso para cada uno de los canales y cuando finaliza. Una vez terminada la calibración se llama a las funciones del código fuente relacionadas con la lectura de los registros de calibración y con aplicar ésta a los valores de las ganancias de tensión y corriente. Así en la Figura 16 se muestra un flujograma con el funcionamiento del flujo del bucle principal.

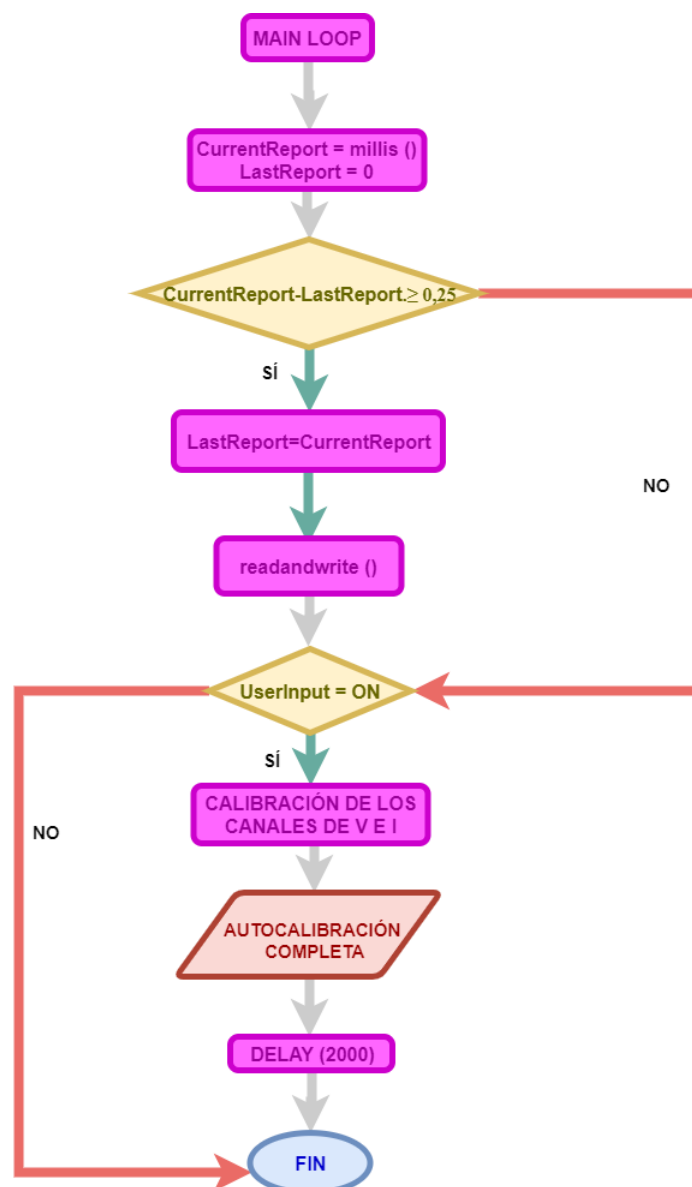


Figura 16: Flujograma programa principal: Main loop

En este archivo también se han cambiado fragmentos de código. En este caso se ha modificado cómo se muestran los datos por pantalla y la frecuencia con la que se muestran, eligiendo un valor de 4 kHz. Los datos se visualizan en columnas, ya que el tenerlos de esta manera en un archivo de texto facilita el procesado posterior de los mismos.

Por último, también se ha cambiado el tipo de alguna variable debido a que alcanzaban valores que producían overflow.

### 5.7.2 DATOS

En este apartado se explica toda la fase de adquisición de datos y las diversas pruebas que se han realizado con la finalidad de obtener datos de los distintos tipos de cargas.

Los datos tanto de los valores eficaces como de las potencias se adquieren a 4 kHz y se muestran en pantalla cada 0,25 ms. Sin embargo, las velocidades de transmisión y de funcionamiento del puerto serie y del SPI están limitadas, lo que significa que el módulo Arduino no es capaz de procesar los datos a la frecuencia de 4 kHz.

Ambas velocidades se han programado para su valor máximo, según el manual de usuario del chip 115200 baudios para el puerto serie y 1 MHz para el SPI. Con estos valores se consiguen unos 76 datos/s.

Aunque el número de muestras que se consiguen no es el deseado, es una buena estimación ya que los contadores actuales son capaces de recoger datos de la red cada 15 minutos y con dispositivos ad-hoc se realiza normalmente 1 dato por segundo. También con el número de datos obtenidos y el algoritmo aplicado se consigue detectar la conexión y desconexión de la mayoría de las cargas.

A su vez, para poder procesar los datos con el algoritmo es necesario que éstos sean almacenados en un fichero de texto o .csv. El IDE de Arduino no dispone de esta funcionalidad por lo que se ha optado por emplear un programa hiperterminal que la incorpore.

De esta manera, se ha elegido emplear el hiperterminal “Realterm”; el cual permite, tras haber configurado los parámetros de funcionamiento correctamente, mostrar los datos por pantalla y capturarlos en un fichero de texto o en un fichero.m [20].



Así, se han tenido que configurar los siguientes parámetros:

1. Puerto: El puerto a través del que se conecta el Arduino, en este caso es el COM6.
2. Velocidad del puerto serie: La misma velocidad a la que se ha programado el chip, 115200 baudios.
3. Formato en el que se reciben los datos: En este caso ANSI.

En la figura 17 se muestra un ejemplo de la interfaz de usuario que incorpora dicho programa.

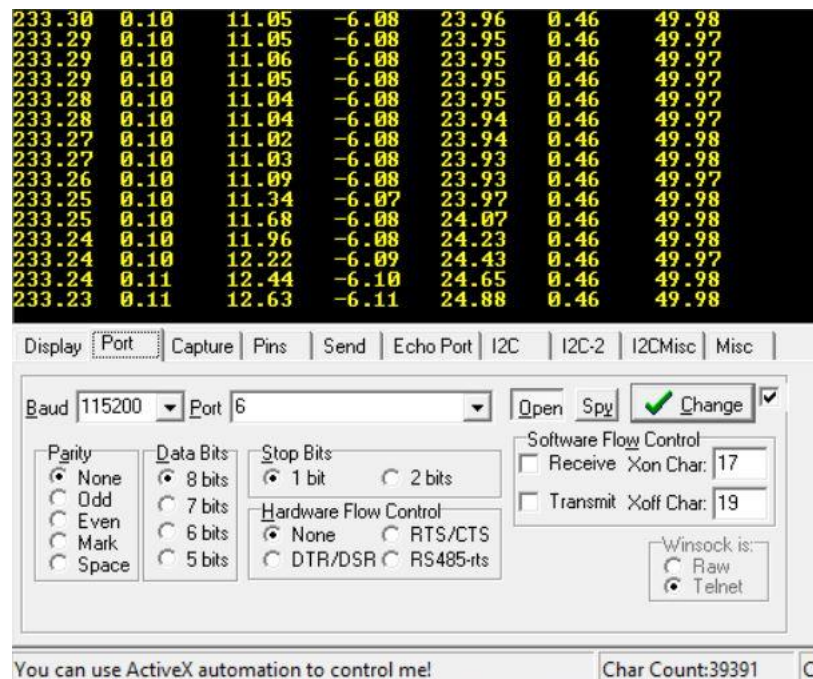


Figura 17: Interfaz Realterm

En cuanto a las pruebas y los datos capturados, se han realizado pruebas con 3 tipos de cargas distintas:

1. Carga resistiva: Bombilla de 40 W y 0,18 A. Sirvió como base para detectar errores en los valores adquiridos ya que es la carga más lineal.
2. Carga inductiva: Secador de 2500 W de potencia activa y un consumo de alrededor a 2,6 A (aunque la mayor parte de la potencia es debida a la resistencia de calefacción, el motor le da un carácter inductivo).
3. Carga capacitiva (realmente una carga no lineal - rectificador a la entrada del cargador): Un ordenador con un consumo de unos 50/60 W al estar casi descargado.

También se han capturado los datos de la red sin conectar ninguna carga.

En el apartado de resultados se muestran las señales tanto de los valores eficaces, como de las potencias y de la detección de eventos para cada una de las tres cargas.

En la figura 18 se muestran los datos capturados con las tres cargas conectadas a la vez.

229.44	0.24	50.26	-8.05	54.51	0.98	49.94
229.44	0.24	50.24	-8.05	54.52	0.98	49.95
229.44	0.24	50.22	-8.05	54.52	0.98	49.95
229.44	0.24	50.21	-8.05	54.52	0.98	49.95
229.44	0.24	50.18	-8.05	54.51	0.98	49.94
229.44	0.24	50.17	-8.05	54.50	0.98	49.94
229.44	0.24	50.15	-8.05	54.49	0.98	49.94
229.44	0.24	50.15	-8.05	54.49	0.98	49.94
229.44	0.24	50.18	-8.06	54.48	0.98	49.94
229.44	0.24	50.21	-8.05	54.48	0.98	49.94
229.44	0.24	50.31	-8.06	54.49	0.98	49.94
229.45	0.24	50.47	-8.06	54.51	0.98	49.94
229.45	0.24	50.83	-8.06	54.58	0.98	49.94
229.45	0.24	51.38	-8.06	54.73	0.98	49.94
229.45	0.24	52.03	-8.07	54.98	0.98	49.94
229.46	0.24	52.86	-8.09	55.34	0.98	49.94
229.46	0.24	53.73	-8.11	55.82	0.98	49.94
229.46	0.25	54.66	-8.14	56.41	0.98	49.94
229.47	0.25	55.65	-8.18	57.11	0.98	49.94
229.47	0.25	56.62	-8.24	57.93	0.98	49.94
229.47	0.26	57.64	-8.30	58.81	0.98	49.94
229.47	0.26	58.60	-8.37	59.76	0.98	49.94
229.48	0.26	59.54	-8.46	60.76	0.98	49.94
229.48	0.27	60.49	-8.55	61.81	0.98	49.94
229.48	0.27	61.37	-8.65	62.88	0.98	49.95
229.49	0.28	62.29	-8.76	63.98	0.98	49.95
229.49	0.28	63.12	-8.87	65.09	0.98	49.94
229.49	0.29	63.93	-8.98	66.20	0.98	49.94
229.50	0.29	64.69	-9.09	67.31	0.98	49.94
229.50	0.30	65.35	-9.21	68.39	0.98	49.95
229.50	0.30	66.01	-9.33	69.44	0.98	49.94

Figura 18: Datos relacionados a la conexión y desconexión de varias cargas

En la figura se pueden apreciar, por lo columnas, los datos relacionados a la tensión y corriente eficaces, potencia activa, reactiva y aparente, factor de potencia y frecuencia relacionados con las conexiones y desconexiones de las distintas cargas.

### 5.7.3 ALGORITMO

En este apartado se explica el algoritmo realizado para detectar las conexiones y desconexiones de las cargas.

Como se ha explicado anteriormente, el algoritmo se ha realizado en Matlab, por lo que se necesitaba que los datos fueran almacenados en un archivo de texto.

El algoritmo se basa, principalmente en detectar los cambios en la señal de corriente eficaz, ya que es en ésta donde reside la mayor información acerca del comportamiento de la carga.

Primeramente, se cargan los datos y se representan las señales de tensión y de corriente. Como información adicional también se pueden representar las distintas potencias, comparando unas cargas con otras.

Posteriormente se deriva la señal de corriente de la carga normalizada para obtener los picos de la señal empleando la función “*diff*”, la cual incluye Matlab en sus librerías. Ésta calcula las diferencias entre los valores de la corriente eficaz que se encuentran próximos entre sí a lo largo de la primera cota de matriz cuyo tamaño es distinto de 1. Esta función también permite que se puedan establecer la dimensión de la cota y el orden de diferencias. [18].

En el presente caso, los resultados que va calculando la función se elevan al cuadrado evitando así, tener una señal con valores negativos de la derivada y poder aplicar el resto del algoritmo.

Para terminar, se detectan los cambios que se dan en la señal de corriente una derivada. Para ello hay que detectar los picos de la misma ya que estos indican si ha habido una conexión o desconexión de cargas.

Para detectar los picos se emplea la función “*findpeaks*” que también incorpora Matlab en sus librerías. Ésta devuelve un vector con los máximos locales del vector de la señal de entrada. También se puede programar para que devuelva, además, los índices en los que se producen los picos y las anchuras y las prominencias de éstos. [19].

El objetivo del presente caso no es que se localicen todos los máximos, sino que se localicen aquellos que representen un cambio en la red. Para ello esta función permite configurar dos parámetros: “*MinPeakDistance*” que es un tiempo de ventana, por el cual se ignoran los picos que estén entre ellos a dicho valor y “*MinPeakHeight*”- que permite detectar aquellos picos que superen un valor determinado. En el presente caso se ha programado una amplitud de  $1 \cdot 10^{-6}$  y una distancia programable dependiendo del tipo de carga y del número de muestras. Estos valores se pueden modificar dependiendo de la carga, sin embargo, gracias a la normalización se han generalizado para los distintos ejemplos.

Por último, se obtienen los resultados representando las señales tanto de corriente eficaz, como la derivada y los picos detectados sobre esta para poder comprobar la eficacia del algoritmo. En el siguiente apartado se exponen los resultados y las conclusiones obtenidas.

En la figura 19 se muestra un diagrama de bloques que representa el funcionamiento del algoritmo descrito.

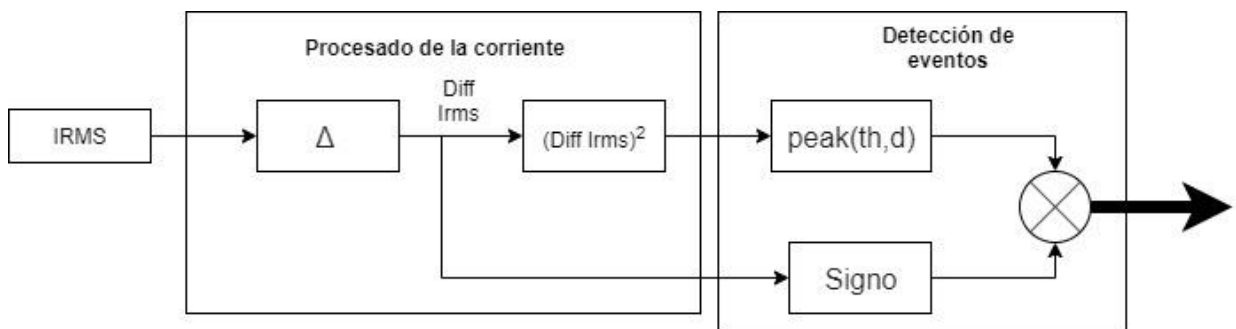


Figura 19: Diagrama de bloques del algoritmo desarrollado

Cabe mencionar que se ha intentado normalizar la señal de corriente eficaz, siguiendo la ecuación 8, pero se ha optado por manejar los valores reales de la corriente. La señal de corriente es muy sensible a los cambios introducidos por las distintas cargas que se conectan a la red, por lo que al normalizarla se pueden generalizar los cálculos para cualquier carga. En este caso se ha intentado normalizarla entre los valores de 0 y 1 ya que la mayoría de los datos están comprendidos entre éstos, sin embargo, los resultados que se obtuvieron no fueron significantes a la hora de aplicar el algoritmo ya que aún así se tenían que modificar los parámetros empleados en la función *findpeaks*.

$$Inorm = \frac{I_{rms}}{I_{max}}$$

Ecuación 8: Corriente normalizada entre 0 y 1.

### 5.7.4 RESULTADOS.

Una vez explicados tanto el código como el algoritmo realizado se exponen los resultados obtenidos tras aplicar la detección de eventos a las cargas mencionadas en los apartados anteriores.

#### 1. CARGA 1: BOMBILLA DE 40 W

En este caso se recogen los resultados tras aplicar el algoritmo a una carga resistiva, habiendo capturado un total de 1025 muestras.

Como se puede observar en la figura 20, la tensión eficaz oscila muy poco, entre los 229,2 y los 230,3, sin embargo, en la corriente se ve como se mantiene a 0 hasta que se conecta la bombilla a la red en el instante 185, donde comienza un transitorio hasta llegar al valor de 0,18 A, que la señal se estabiliza hasta que la bombilla se desconecta.

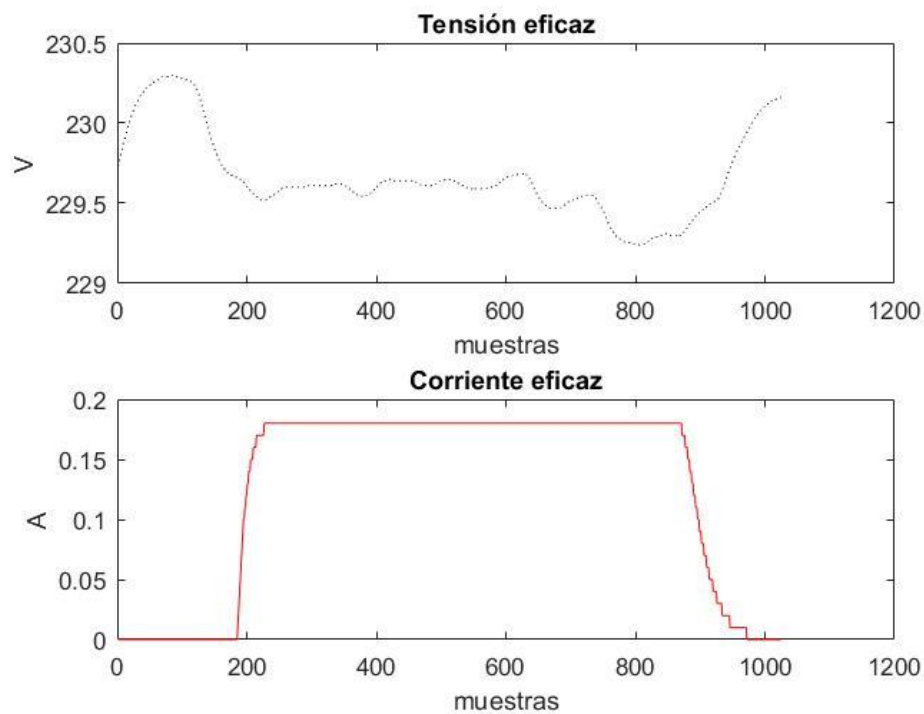


Figura 20: Tensión y corrientes eficaces de una bombilla de 40 W

Las potencias activa, aparente y reactiva se representan en la figura 21, se puede observar cómo se elevan las potencias activa y aparente tomando prácticamente los mismos valores, ya que es una carga resistiva y como la potencia reactiva apenas toma valores.

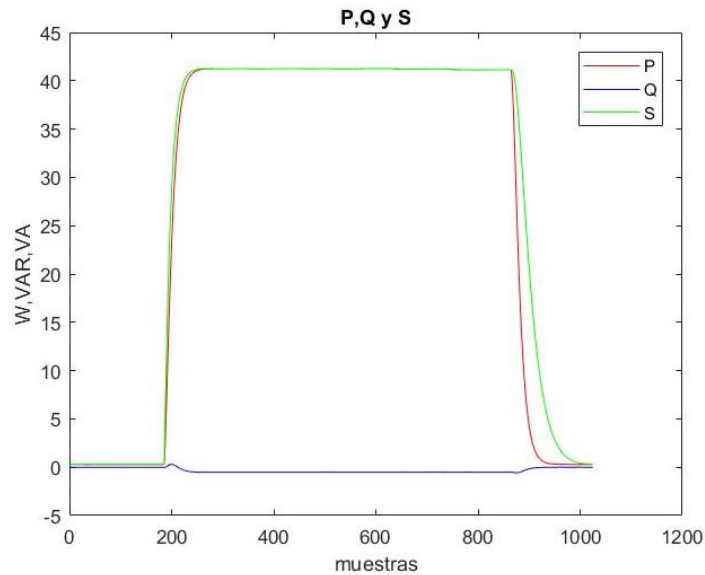


Figura 21: Potencias activa, reactiva y aparente de una bombilla de 40W

Por último, se muestran los resultados tras aplicar el algoritmo al valor obtenido que se obtiene tras derivar la señal de la corriente. Para obtener los picos en la función “*findpeaks*” se han programado los siguientes valores:

1. MinPeakHeight =  $1 \cdot 10^{-4}$
2. MinPeakdistance = 500.

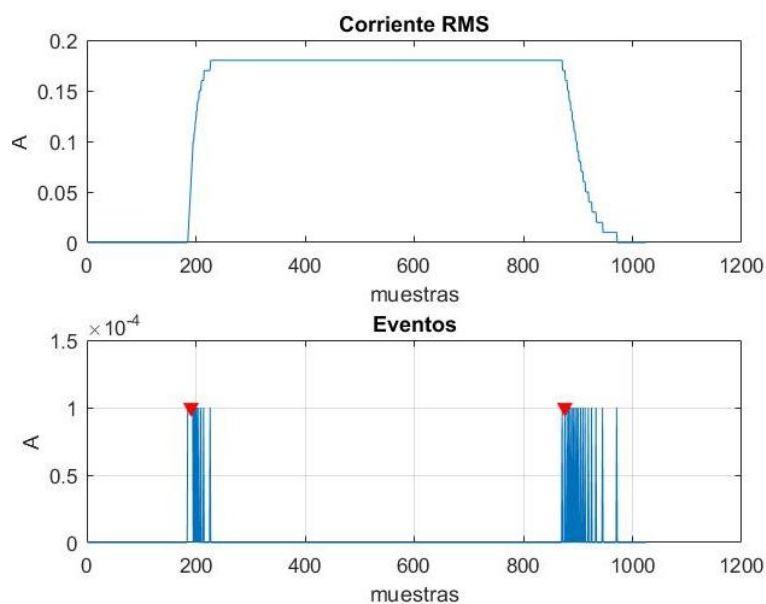


Figura 22: Eventos de una bombilla de 40W

Como se puede apreciar, en la señal derivada se obtienen varios picos debidos al transitorio de la señal, sin embargo, gracias al ajuste en los parámetros de la función “*findpeaks*” se puede detectar el momento preciso en el que se ha conectado o desconectado la carga.

## 2. CARGA 2: SECADOR DE 2500W.

A continuación, se muestran los datos y los resultados obtenidos tras aplicar el algoritmo de detección de eventos a una carga inductiva, la cual tiene distintos modos de funcionamiento. El número de muestras a evaluar es de 1472.

En la figura 23 se pueden observar los valores que toman la corriente y la tensión eficaz tanto en el momento de conexión y desconexión de la carga como cuando se activan los distintos modos de funcionamiento de ésta.

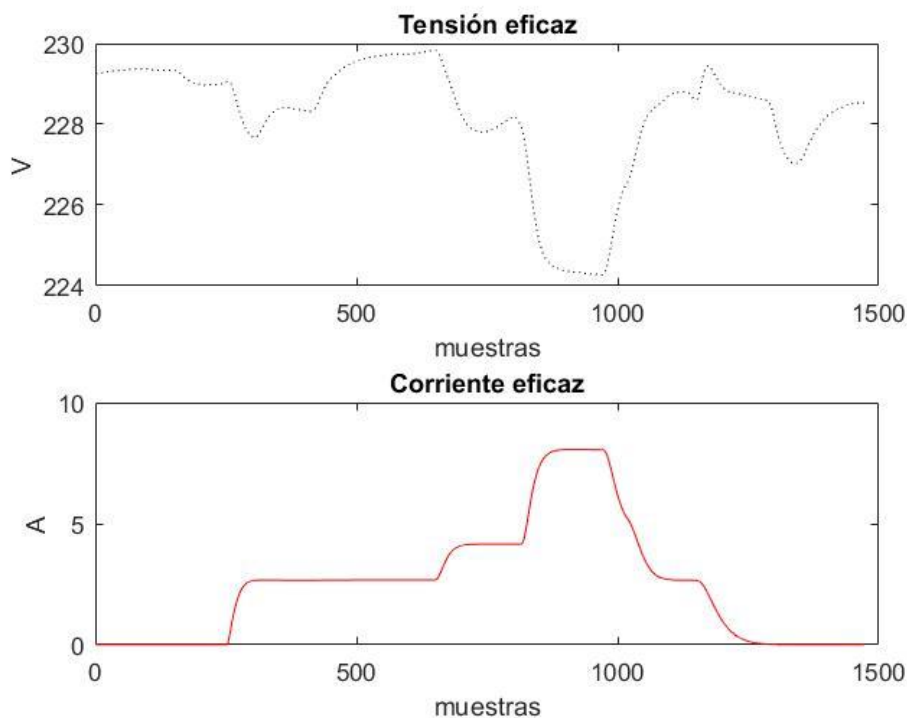


Figura 23: Tensión y corriente eficaces de un secador de 2500 W

Como se puede observar en la figura el transitorio de la corriente es muy lineal, pero con unos picos muy pronunciados, lo que es de ayuda a la hora de aplicar el algoritmo de detección de eventos. En cuanto a la tensión se ve cómo la corriente aumenta hasta casi los 10 A ésta decae hasta casi los 224 V.

En la figura 24 se representan las potencias obtenidas, en ésta se puede ver como la potencia reactiva es pequeña en relación con la activa, sobre todo cuando el calentamiento es fuerte (alta potencia activa). El efecto inductivo se percibe más, lógicamente, cuando el calentamiento es menor ( $S > P$  claramente).

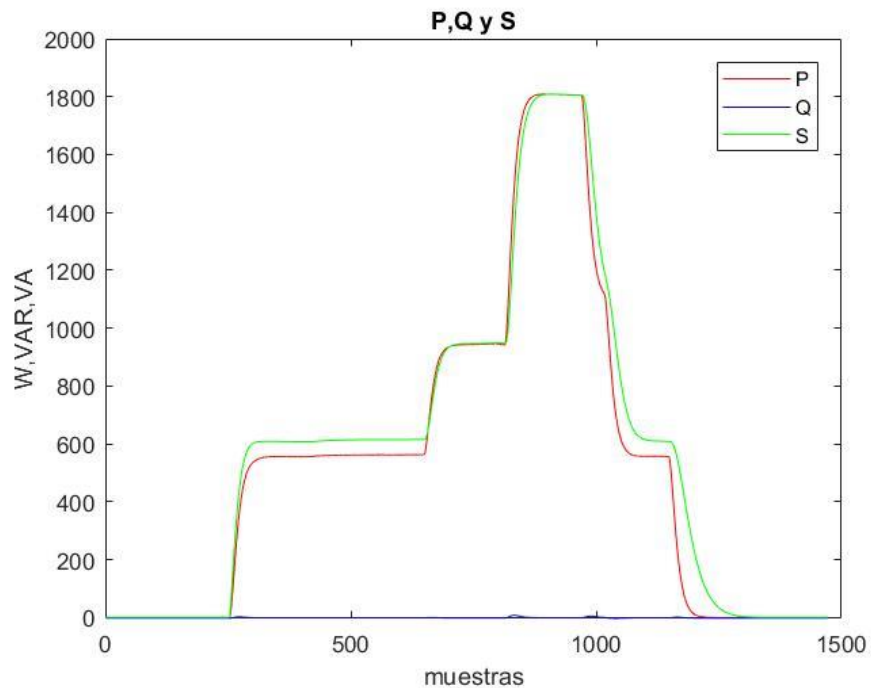


Figura 24: Potencias activa, reactiva y aparente de un secador de 2000 W



En la figura 25 se muestra el resultado de aplicar el algoritmo de detección de eventos a la señal de corriente derivada. Como se puede observar se detectan todos los cambios en el modo de funcionamiento que han tenido lugar en las 1472 muestras recogidas. Para poder detectar los eventos en la función *findpeaks* se han programado los siguientes parámetros:

1.  $\text{MinPeakHeight} = 1 \cdot 10^{-3}$
2.  $\text{MinPeakdistance} = 50$ .

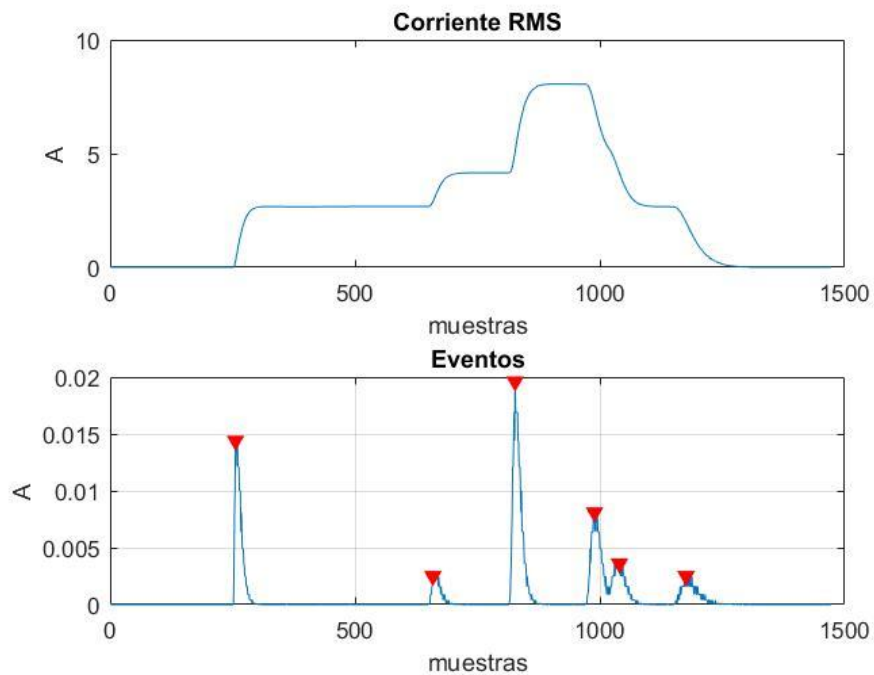


Figura 25: Eventos de un secador de 2500 W

### 3. CARGA 3: ORDENADOR CON UN CONSUMO DE 50/60 W

En el presente apartado se muestran los resultados obtenidos en una carga no lineal (rectificador con carga de tipo capacitiva), a lo largo de 2655 muestras.

La tensión y corrientes aparentes se representan en la figura 26. Como se puede observar, la corriente presenta unos valores muy poco estables, propios de este tipo de cargas con corrientes pulsadas (modo discontinuo), lo que dificulta la aplicación del algoritmo como se verá posteriormente. En cuanto la tensión se mantiene más o menos estable menos con la caída a 231 V que tiene lugar cuando la carga se desconecta

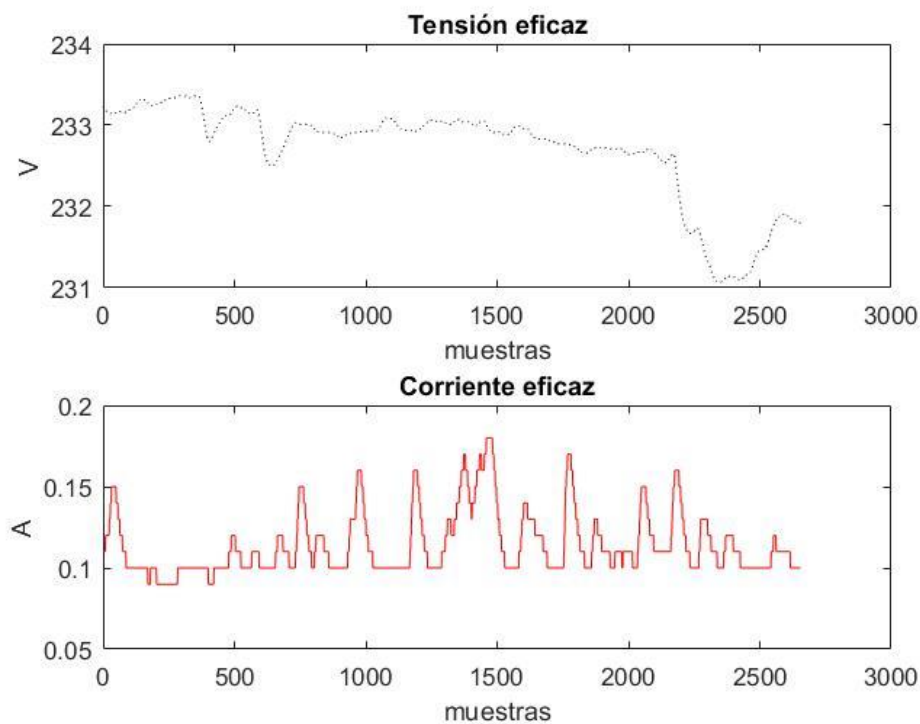


Figura 26: Tensión y corriente eficaces de una carga no lineal

En la figura 27 se representan las potencias, como se puede observar tiene un comportamiento muy distinto al de las dos cargas anteriores. En este caso las potencias activa y aparente tienen la misma forma, pero la aparente siempre es algo superior que la activa y como se puede apreciar en esta ocasión se tienen valores negativos de potencia reactiva (efecto capacitivo). Se puede observar también que los valores de potencia activa no superan los 40 W en la mayoría de las muestras lo que quiere decir es que el ordenador no estaba descargado por completo cuando se conectó a la red.

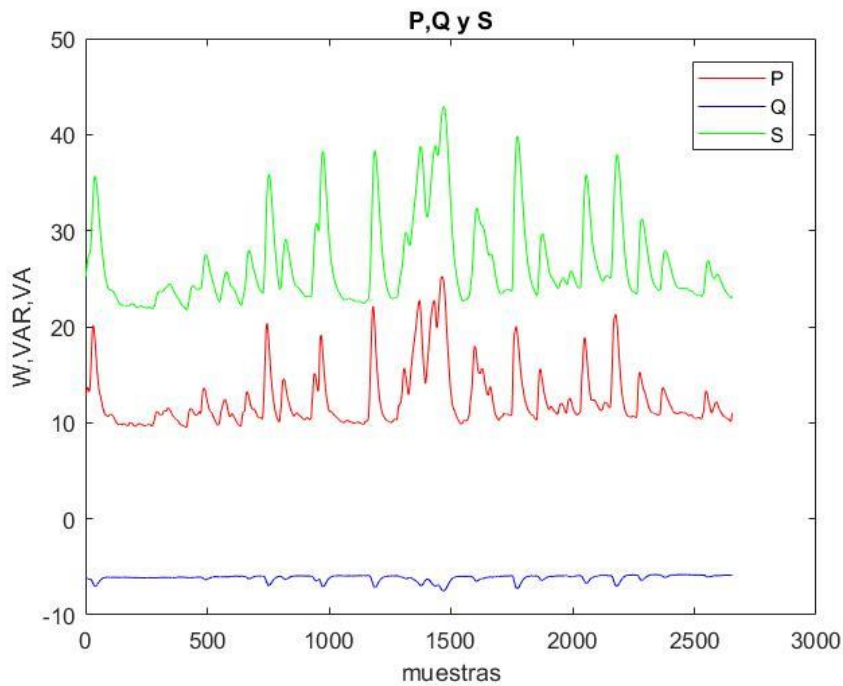


Figura 27: Potencias activa, reactiva y aparente de una carga no lineal.

Por último, se representa la detección de eventos con los siguientes valores para la función *findpeaks*

1.  $\text{MinPeakHeight} = 1 \cdot 10^{-5}$
2.  $\text{MinPeakDistance} = 50$ .

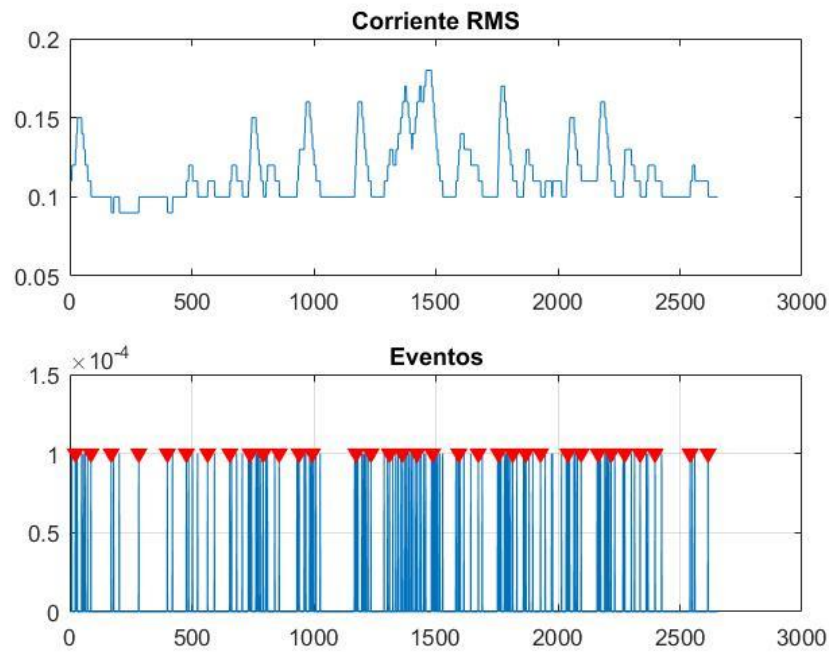


Figura 28: Eventos de una carga no lineal.

Como se puede observar, en este caso el algoritmo de detección de eventos detecta todos los cambios que tiene la corriente en este tipo de cargas, aunque no sean conexiones y/o desconexiones a la red, por lo que habría que proponer otro tipo de algoritmo específico para este tipo de cargas.

#### 4. CARGA 4: CONEXIONES Y DESCONEXIONES DE LAS CARGAS ANTERIORES

Por último, se muestran los resultados recogidos al combinar las conexiones y desconexiones de las cargas descritas anteriormente juntas. Se han recogido 22226 muestras siguiendo la siguiente secuencia de conexiones y desconexiones:

1. Enchufar secador
2. Encender secador a potencia media
3. Apagar secador.
4. Encender bombilla.
5. Conectar ordenador.
6. Desconectar ordenador.
7. Apagar bombilla.

Así, en la figura 29 se puede observar como en la corriente se reflejan claramente las transiciones bruscas de nivel, indicando las conexiones y desconexiones de cada una de las cargas. Por ejemplo, la parte más estable es cuando se conecta la carga resistiva. En cuanto a la tensión se puede ver como fluctúa bastante más que en los casos anteriores debido a las conexiones de las 3 cargas, debido a los picos de corriente

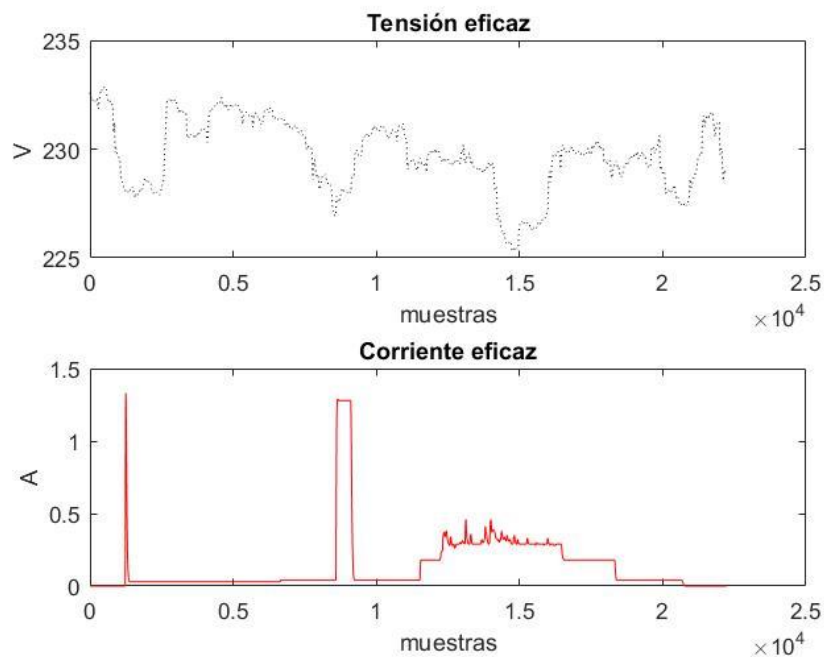


Figura 29: Tensión y corriente eficaz de varias cargas

En la figura 30 se observan las potencias, así la aparente es sensiblemente superior a la activa y se aprecia como en el momento que se conecta el ordenador aparecen los valores negativos de la potencia reactiva.

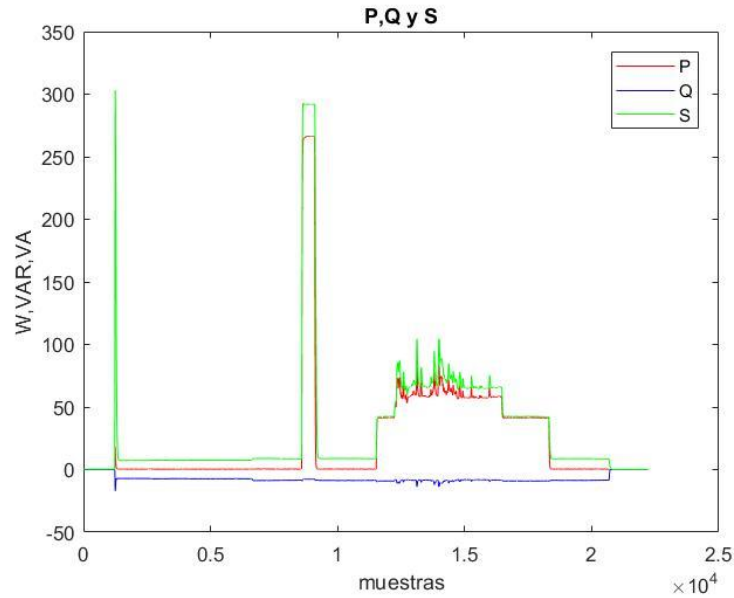


Figura 30: Potencias activa, reactiva y aparente de varias cargas

Por último, en la figura 31 se observa la detección de eventos. En ésta se ve como el algoritmo detecta todas las transiciones de nivel bruscas que indican una conexión o una desconexión a excepción de cuando se conecta la carga no lineal que detecta el ruido que produce dicha carga.

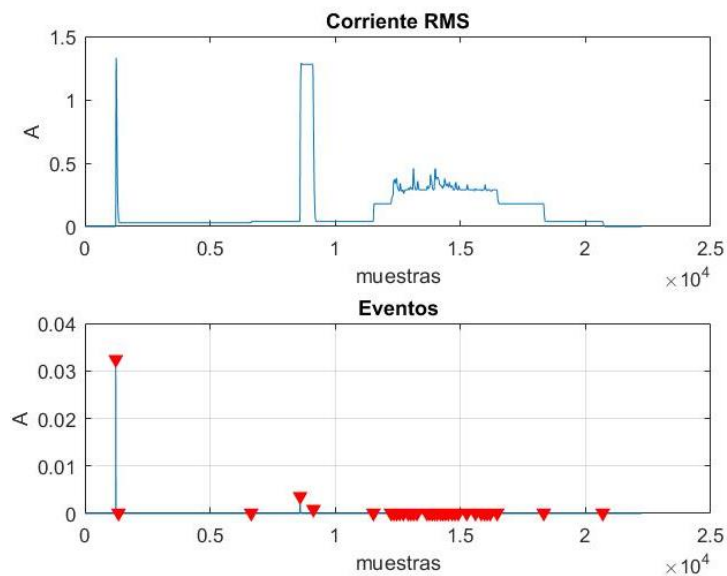


Figura 31: Eventos de varias cargas

## 5.8 CONCLUSIONES Y TRABAJOS FUTUROS

En el presente apartado se recogen las conclusiones obtenidas acerca de los resultados que se han recogido y los posibles trabajos futuros como mejoras que se pueden desarrollar a partir de este TFG.

La tarjeta de adquisición de datos empleada tiene múltiples funcionalidades que se han descrito anteriormente como por ejemplo que internamente es capaz de procesar datos a una frecuencia de muestreo de 4kHz, se pueden adquirir directamente los valores eficaces de las cargas o su versatilidad ya que está preparada para poder conectar un módulo Arduino y poder procesar los datos. También el fabricante proporciona un código de adquisición lo que facilita al usuario poder entender el funcionamiento del chip y sus funcionalidades.

En este TFG se han intentado aprovechar todas las funcionalidades del chip, consiguiendo los siguientes objetivos:

1. Realizar un estudio del funcionamiento y las características de la tarjeta y del chip a partir de toda la información proporcionada por el fabricante.
2. Aprovechando todas las características y funcionalidades de la tarjeta y aplicando mejoras en el código proporcionado por el fabricante, se ha conseguido obtener los valores eficaces de tensión y corriente, los de las potencias activa, reactiva y aparente, el factor de potencia y la frecuencia de la red.
3. Obtener dichos datos a una frecuencia de unas 76 muestras por segundo.
4. Adquirir datos de distintas cargas y aplicar un algoritmo capaz de detectar las conexiones y desconexiones de las mismas en la mayoría de los casos.

Así, se pueden realizar trabajos futuros que sigan la línea de trabajo del presente TFG y completen el trabajo desarrollado hasta aquí.

En cuanto a las características de la tarjeta, como se ha especificado en ocasiones anteriores su frecuencia de adquisición de los datos instantáneos, así como el procesado de estos para obtener el resto de los valores a partir de ellos es de 4 kHz. Sin embargo, las limitaciones debidas a las velocidades de transmisión y funcionamiento del puerto serie y del SPI no permiten que los datos puedan ser procesados por el módulo Arduino a esa frecuencia, llegando a conseguir 76 datos/s. Este aspecto se podría mejorar trabajando directamente desde la tarjeta de adquisición o empleando un microprocesador con unas características más potentes que las que ofrece el módulo de Arduino.

A su vez, también, se pueden adquirir los datos de corriente y tensión instantáneos, trabajar sobre ellos para conseguir los valores eficaces y las distintas potencias y aplicar distintos algoritmos de detección de eventos sobre los mismos.

Respecto al algoritmo, tal y como se recoge en la sección de resultados el algoritmo realizado no logra detectar la conexión y desconexión de cargas capacitivas las cuales incluyen varios picos debido a las cargas y descargas del condensador. Así, se puede desarrollar un algoritmo de detección de eventos particularizado para este tipo de cargas.

Por último, como complemento a este TFG se puede desarrollar un algoritmo clasificador el cual, una vez detectados los eventos en la red pueda clasificar el tipo de carga que ha realizado el evento, basándose en el comportamiento de ésta y en su estudio por separado.

## 6. CÓDIGO Y ALGORITMO

A continuación, se recoge tanto el código empleado para la adquisición de datos como el código realizado para la detección de eventos en Matlab.

### 6.1 CÓDIGO PARA LA ADQUISICIÓN DE DATOS.

En este apartado se recoge el código relacionado con la adquisición de datos: El código fuente y el programa principal.

#### 6.1.1 ADE9153AAPI.cpp

```
#include <Arduino.h>
#include <SPI.h>
#include "ADE9153AAPI.h"
#include <Wire.h>

ADE9153AClass::ADE9153AClass ()
{
}

void ADE9153AClass::SetupADE9153A(void)
{
    SPI_Write_16(REG_AI_PGAGAIN,ADE9153A_AI_PGAGAIN);
    SPI_Write_32(REG_CONFIG0,ADE9153A_CONFIG0);
    SPI_Write_16(REG_CONFIG1,ADE9153A_CONFIG1);
    SPI_Write_16(REG_CONFIG2,ADE9153A_CONFIG2);
    SPI_Write_16(REG_CONFIG3,ADE9153A_CONFIG3);
    SPI_Write_16(REG_ACCMODE,ADE9153A_ACCMODE);
    SPI_Write_32(REG_VLEVEL,ADE9153A_VLEVEL);
    SPI_Write_16(REG_ZX_CFG,ADE9153A_ZX_CFG);
    SPI_Write_32(REG_MASK,ADE9153A_MASK);
    SPI_Write_32(REG_ACT_NL_LVL,ADE9153A_ACT_NL_LVL);
    SPI_Write_32(REG_REACT_NL_LVL,ADE9153A_REACT_NL_LVL);
    SPI_Write_32(REG_APP_NL_LVL,ADE9153A_APP_NL_LVL);
    SPI_Write_16(REG_COMPMODE,ADE9153A_COMPMODE);
    SPI_Write_32(REG_VDIV_RSMALL,ADE9153A_VDIV_RSMALL);
    SPI_Write_16(REG_EP_CFG,ADE9153A_EP_CFG);
    SPI_Write_16(REG_EGY_TIME,ADE9153A_EGY_TIME); //Energy
    accumulation ON
    SPI_Write_16(REG_TEMP_CFG,ADE9153A_TEMP_CFG);
}
```



```

bool ADE9153AClass::SPI_Init(uint32_t SPI_speed , uint8_t
chipSelect_Pin)
{
    digitalWrite(chipSelect_Pin, HIGH);          //Set Chip select
pin high
    pinMode(chipSelect_Pin, OUTPUT);            //Set Chip select
pin as output
    SPI.begin();                                //Initiate SPI port
    SPI.beginTransaction(SPISettings(SPI_speed,MSBFIRST,SPI_MODE0))
;
    //Setup SPI parameters

    _chipSelect_Pin = chipSelect_Pin;

    SPI_Write_16(REG_RUN,ADE9153A_RUN_ON);
    delay(100);
    if (SPI_Read_32(REG_VERSION_PRODUCT) != 0x0009153A)
        return false;

    return true;
}

void ADE9153AClass:: SPI_Write_16(uint16_t Address , uint16_t Data )
{
    uint16_t temp_address;

    digitalWrite(_chipSelect_Pin, LOW);
    temp_address = ((Address << 4) & 0xFFFF0);    //shift
address to align with cmd packet
    SPI.transfer16(temp_address);
    SPI.transfer16(Data);

    digitalWrite(_chipSelect_Pin, HIGH);
}

void ADE9153AClass:: SPI_Write_32(uint16_t Address , uint32_t Data )
{
    uint16_t temp_address;
    uint16_t temp_highpacket;
    uint16_t temp_lowpacket;

    temp_highpacket= (Data & 0xFFFFF0000)>>16;
    temp_lowpacket= (Data & 0x0000FFFF);

    digitalWrite(_chipSelect_Pin, LOW);

    temp_address = ((Address << 4) & 0xFFFF0);    //shift
address to align with cmd packet
    SPI.transfer16(temp_address);
    SPI.transfer16(temp_highpacket);
    SPI.transfer16(temp_lowpacket);

    digitalWrite(_chipSelect_Pin, HIGH);
}

```

```

uint16_t ADE9153AClass:: SPI_Read_16(uint16_t Address)
{
    uint16_t temp_address;
    uint16_t returnData;

    digitalWrite(_chipSelect_Pin, LOW);

    temp_address = (((Address << 4) & 0xFFFF0)+8);
    SPI.transfer16(temp_address);
    returnData = SPI.transfer16(0);

    digitalWrite(_chipSelect_Pin, HIGH);
    return returnData;
}

uint32_t ADE9153AClass:: SPI_Read_32(uint16_t Address)
{
    uint16_t temp_address;
    uint32_t temp_highpacket;
    uint16_t temp_lowpacket;
    uint32_t returnData;

    digitalWrite(_chipSelect_Pin, LOW);

    temp_address = (((Address << 4) & 0xFFFF0)+8);
    SPI.transfer16(temp_address);
    temp_highpacket = SPI.transfer16(0);
    temp_lowpacket = SPI.transfer16(0);

    digitalWrite(_chipSelect_Pin, HIGH);

    returnData = temp_highpacket << 16;
    returnData = returnData + temp_lowpacket;

    return returnData;
}

void ADE9153AClass:: ReadEnergyRegs(EnergyRegs *Data)
{
    int32_t tempReg;
    float tempValue;

    tempReg = int32_t (SPI_Read_32(REG_AWATTHR_HI));
    Data->ActiveEnergyReg = tempReg;
    tempValue = (float)tempReg * CAL_ENERGY_CC / 1000;
    Data->ActiveEnergyValue = tempValue;
    //Energy in mWhr

    tempReg = int32_t (SPI_Read_32(REG_AFVARHR_HI));
    Data->FundReactiveEnergyReg = tempReg;
    tempValue = (float)tempReg * CAL_ENERGY_CC / 1000;
    Data->FundReactiveEnergyValue = tempValue;
    //Energy in mVARhr

    tempReg = int32_t (SPI_Read_32(REG_AVAHR_HI));
    Data->ApparentEnergyReg = tempReg;
    tempValue = (float)tempReg * CAL_ENERGY_CC / 1000;
    Data->ApparentEnergyValue = tempValue;
    //Energy in mVAhr
}

```

```

void ADE9153AClass:: ReadPowerRegs (PowerRegs *Data)
{
    int32_t tempReg;
    float tempValue;

    tempReg = int32_t (SPI_Read_32 (REG_AWATT));
    Data->ActivePowerReg = tempReg;
    tempValue = (float)tempReg * CAL_POWER_CC / 1000;
    Data->ActivePowerValue =
tempValue; //Power in mW

    tempReg = int32_t (SPI_Read_32 (REG_AFVAR));
    Data->FundReactivePowerReg = tempReg;
    tempValue = (float)tempReg * CAL_POWER_CC / 1000;
    Data->FundReactivePowerValue = tempValue;
    //Power in mVAR

    tempReg = int32_t (SPI_Read_32 (REG_AVA));
    Data->ApparentPowerReg = tempReg;
    tempValue = (float)tempReg * CAL_POWER_CC / 1000;
    Data->ApparentPowerValue = tempValue;
    //Power in mVA
}

void ADE9153AClass:: ReadRMSRegs (RMSRegs *Data)
{
    uint32_t tempReg;
    float tempValue;

    tempReg = int32_t (SPI_Read_32 (REG_AIRMS));
    Data->CurrentRMSReg = tempReg;
    tempValue = (float)tempReg * CAL_IRMS_CC / 1000; //RMS in
mA
    Data->CurrentRMSValue = tempValue;

    tempReg = int32_t (SPI_Read_32 (REG_AVRMS));
    Data->VoltageRMSReg = tempReg;
    tempValue = (float)tempReg * CAL_VRMS_CC / 1000; //RMS in
mV
    Data->VoltageRMSValue = tempValue;
}

void ADE9153AClass:: ReadHalfRMSRegs (HalfRMSRegs *Data)
{
    uint32_t tempReg;
    float tempValue;

    tempReg = int32_t (SPI_Read_32 (REG_AIRMS_OC));
    Data->HalfCurrentRMSReg = tempReg;
    tempValue = (float)tempReg * CAL_IRMS_CC / 1000; //Half-
RMS in mA
    Data->HalfCurrentRMSValue = tempValue;

    tempReg = int32_t (SPI_Read_32 (REG_AVRMS_OC));
    Data->HalfVoltageRMSReg = tempReg;
    tempValue = (float)tempReg * CAL_VRMS_CC / 1000; //Half-
RMS in mV
    Data->HalfVoltageRMSValue = tempValue;
}

```

```

void ADE9153AClass:: ReadPQRegs (PQRegs *Data)
{
    int32_t tempReg;
    uint16_t temp;
    float mulConstant;
    float tempValue;

    tempReg=int32_t (SPI_Read_32(REG_APF)); //Read PF register
    Data->PowerFactorReg = tempReg;
    tempValue=(float)tempReg/(float)134217728; //Calculate PF
    Data->PowerFactorValue=tempValue;

    tempReg=int32_t (SPI_Read_32(REG_APERIOD)); //Read PERIOD
register
    Data->PeriodReg = tempReg;
    tempValue=(float) (4000*65536)/(float) (tempReg+1); //Calculate
Frequency
    Data->FrequencyValue = tempValue;

    temp=SPI_Read_16(REG_ACCMODE); //Read frequency setting
register
    if((temp&0x0010)>0)
    {
        mulConstant=0.02109375; //multiplier constant
for 60Hz system
    }
    else
    {
        mulConstant=0.017578125; //multiplier constant
for 50Hz system
    }

    tempReg=int16_t (SPI_Read_16(REG_ANGL_AV_AI)); //Read ANGLE
register
    Data->AngleReg_AV_AI=tempReg;
    tempValue=tempReg*mulConstant; //Calculate Angle in degrees
    Data->AngleValue_AV_AI=tempValue;
}

```

```

void ADE9153AClass:: ReadAcalRegs (AcalRegs *Data)
{
    uint32_t tempReg;
    float tempValue;

    tempReg=int32_t (SPI_Read_32 (REG_MS_ACAL_AICC)); //Read AICC
register
Data->AcalAICCReg = tempReg;
tempValue=(float) tempReg/ (float) 2048; //Calculate Conversion
Constant (CC)
Data->AICC=tempValue;
tempReg=int32_t (SPI_Read_32 (REG_MS_ACAL_AICERT)); //Read
AICERT register
Data->AcalAICERTReg = tempReg;

tempReg=int32_t (SPI_Read_32 (REG_MS_ACAL_AVCC)); //Read AVCC
register
Data->AcalAVCCReg = tempReg;
tempValue=(float) tempReg/ (float) 2048; //Calculate Conversion
Constant (CC)
Data->AVCC=tempValue;
tempReg=int32_t (SPI_Read_32 (REG_MS_ACAL_AVCERT)); //Read
AICERT register
Data->AcalAVCERTReg = tempReg;
}

bool ADE9153AClass::StartAcal_AINormal(void)
{
    uint32_t ready = 0;
    int waitTime = 0;

    //ready =
SPI_Read_32 (REG_MS_STATUS_CURRENT); //Read
system ready bit

    while((ready&0x00000001)==0)
    {
        ready = SPI_Read_32 (REG_MS_STATUS_CURRENT);
        if(waitTime>11)
        {
            return false;
        }
        delay(100);
        waitTime++;
    }

    SPI_Write_32 (REG_MS_ACAL_CFG, 0x00000013);
    return true;
}

```

```

bool ADE9153AClass::StartAcal_AITurbo(void)
{
    uint32_t ready = 0;
    int waitTime = 0;

    while((ready&0x00000001)==0)
    {
        ready = SPI_Read_32(REG_MS_STATUS_CURRENT);
        //Read system ready bit
        if(waitTime>15)
        {
            return false;
        }
        delay(100);
        waitTime++;
    }

    SPI_Write_32(REG_MS_ACAL_CFG, 0x00000017);
    return true;
}

bool ADE9153AClass::StartAcal_AV(void)
{
    uint32_t ready =0;
    int waitTime = 0;

    while((ready&0x00000001)==0)
    {
        ready = SPI_Read_32(REG_MS_STATUS_CURRENT);
        //Read system ready bit
        if(waitTime>15)
        {
            return false;
        }
        delay(100);
        waitTime++;
    }

    SPI_Write_32(REG_MS_ACAL_CFG, 0x00000043);
    return true;
}

void ADE9153AClass::StopAcal(void)
{
    SPI_Write_32(REG_MS_ACAL_CFG, 0x00000000);
}

bool ADE9153AClass::ApplyAcal(float AICC, float AVCC)
{
    int32_t AIGAIN;
    int32_t AVGAIN;

    AIGAIN = (- (AICC / (CAL_IRMS_CC*1000)) - 1) * 134217728;
    AVGAIN = (AVCC / (CAL_VRMS_CC*1000) - 1) * 134217728;

    SPI_Write_32(REG_AIGAIN, AIGAIN);
    SPI_Write_32(REG_AVGAIN, AVGAIN);
}

/*

```

```

void ADE9153AClass:: ReadTemperature(Temperature *Data)
{
    uint32_t trim;
    uint16_t gain;
    uint16_t offset;
    uint16_t tempReg;
    float tempValue;

    SPI_Write_16(REG_TEMP_CFG,ADE9153A_TEMP_CFG);//Start
temperature acquisition cycle
    delay(10); //delay of 2ms. Increase delay if TEMP_TIME is
changed

    trim = SPI_Read_32(REG_TEMP_TRIM);
    gain= (trim & 0xFFFF); //Extract 16 LSB
    offset= ((trim>>16)&0xFFFF); //Extract 16 MSB
    tempReg= SPI_Read_16(REG_TEMP_RSLT); //Read Temperature result
register
    tempValue= ((float)offset / 32.00)-
((float)tempReg*(float)gain/(float)131072);

    Data->TemperatureReg=tempReg;
    Data->TemperatureVal=tempValue;
}

```

### 6.1.2 ADE9153AAPI\_Test.ino

```

#define ARM_MATH_CM0PLUS

#include < SPI . h >
#include < ADE9153A . h >
#include < ADE9153AAPI . h >

/ * Inicializaciones básicas * /
ADE9153AClass ade9153A ;
#define SPI_SPEED 1000000 // Velocidad SPI
#define CS_PIN 8 // 8 -> Arduino Zero. 15 -> ESP8266
#define ADE9153A_RESET_PIN 4 // Pin de reinicio
integrado #define USER_INPUT 5 // Pin del botón de entrada
de usuario integrado
#define LED 6 // Pin de LED integrado

EnergyRegs energyVals ; // Los valores del registro de energía se
leen y almacenan en la estructura
EnergyRegs PowerRegs powerVals ; // Se puede acceder a los datos
de metrología desde estas estructuras
RMSRegs rmsVals ;
PQRegs pqVals ;
AcalRegs acalVals ;
Temperature tempVal ;

void readandwrite (void) ;
void resetADE9153A ( void ) ;

int ledState = LOW ;
int inputState = LOW ;
unsigned long lastReport = 0 ;
const long reportInterval = 0.25 ;
const long blinkInterval = 500 ;

```

```

void setup ( ) {
  /* Configuración del pin y del monitor serie */
  pinMode ( LED , OUTPUT ) ;
  pinMode ( USER_INPUT , INPUT ) ;
  pinMode ( ADE9153A_RESET_PIN , OUTPUT ) ;
  digitalWrite ( ADE9153A_RESET_PIN , HIGH ) ;
  Serial. begin ( 115200 ) ;

  resetADE9153A ( ) ; // reset
  delay ( 1000 ) ;
  /* Inicialización y prueba de SPI */
  bool commscheck = ade9153A . SPI_Init ( SPI_SPEED , CS_PIN ) ;
  // Inicializa SPI
  if ( ! Commscheck ) {
    Serial . println ( "ADE9153A Shield not detected. Plug in Shield
and reset the Arduino" ) ;
    while ( ! commscheck ) { // Espera hasta que arduino se
restablezca
      delay ( 1000 ) ;
    }
  }

  ade9153A . SetupADE9153A ( ) ; // Configurar ADE9153A de acuerdo
con ADE9153AAPI.h
  delay ( 500 ) ;
}

void loop ( ) {

  unsigned long currentReport = millis ( ) ;

  if ((currentReport - lastReport) >= reportInterval){
    lastReport = currentReport;
    readandwrite();
  }

  inputState = digitalRead(USER_INPUT);

  if (inputState == LOW) {
    Serial.println("Autocalibrating Current Channel");
    ade9153A.StartAcal_AINormal();
    runLength(20);
    ade9153A.StopAcal();
    Serial.println("Autocalibrating Voltage Channel");
    ade9153A.StartAcal_AV();
    runLength(40);
    ade9153A.StopAcal();
    delay(100);

    ade9153A.ReadAcalRegs(&acalVals);
    ade9153A.ApplyAcal(acalVals.AICC, acalVals.AVCC);

    Serial.println("Autocalibration Complete");
    delay(2000);
  }
}

```



```

void readandwrite()
{
    /*Read and Print Specific Register using ADE9153A SPI Library */
    //Serial.println(ade9153A.SPI_Read_32(REG_AIRMS),HEX); // AIRMS

    /*Read and Print WATT Register using ADE9153A Read Library*/
    ade9153A.ReadPowerRegs(&powerVals); //Template to read Power
    registers from ADE9153A and store data in Arduino MCU
    ade9153A.ReadRMSRegs(&rmsVals);
    ade9153A.ReadPQRegs(&pqVals);
    ade9153A.ReadTemperature(&tempVal);

    //Datos en columnas, script Matlab
    Serial.print(rmsVals.VoltageRMSValue/1000); //Tensión
    Serial.print("\t");
    Serial.print(rmsVals.CurrentRMSValue/1000); //Corriente
    Serial.print("\t");
    Serial.print(powerVals.ActivePowerValue/1000); //Potencia Activa
    Serial.print("\t");
    Serial.print(powerVals.FundReactivePowerValue/1000); //Potencia
Reactiva
    Serial.print("\t");
    Serial.print(powerVals.ApparentPowerValue/1000); //Potencia
Aparente
    Serial.print("\t");
    Serial.print(pqVals.PowerFactorValue); //PF
    Serial.print("\t");
    Serial.println(pqVals.FrequencyValue); //FREQ.
}

void resetADE9153A(void)
{
    digitalWrite(ADE9153A_RESET_PIN, LOW);
    delay(100);
    digitalWrite(ADE9153A_RESET_PIN, HIGH);
    delay(1000);
    Serial.println("Reset Done");
}

void runLength(unsigned long seconds)
{
    unsigned long startTime = millis();

    while (millis() - startTime < (seconds*1000)){
        digitalWrite(LED, HIGH);
        delay(blinkInterval);
        digitalWrite(LED, LOW);
        delay(blinkInterval);
    }
}

```

### 6.1.3 ALGORITMO

```

%Algoritmo para detectar los eventos que se producen a
partir de los
%valores eficaces

clear all,
close all,
clc,

%Cargar datos de corriente y tensión
load('datos6.txt');

%Mostrar en la misma gráfica tensión y corriente.
subplot(2,1,1);
plot(datos6(:,1),'k:');
xlabel('muestras');
ylabel('V')
title('Tensión eficaz')
subplot(2,1,2);
figure(1)
plot(datos6(:,2),'r');
title('Corriente eficaz');
xlabel('muestras')
ylabel('A')
figure(2)

%Gráficas de las potencias.
plot(datos6(:,3),'r');
hold on;
plot(datos6(:,4),'b');
plot(datos6(:,5),'g');
hold off;
title('P,Q y S');
legend('P','Q','S');
xlabel('muestras')
ylabel('W,VAR,VA')
figure(3);

%Derivar medidas: diff
Irms=datos6(:,2);
drms=(diff(Irms).^2); %Para que no salga negativo

```

```
%Encontrar picos de la señal: findpeaks
%BOMBILLA: 500 y 0,0001.
%SECADOR 50 y 0,001.
%ORDENADOR 50 y 0,00001
%TODO 100 y 0,00001
distance=100; %Diferencias entre conexión y
desconexión
th=1*10^(-5); %Valor de la derivada de los picos.
[pks,locs]=findpeaks(drms,'MinPeakHeight',th,'MinPeakd
istance',distance);

%Visualizar resultados.
subplot (2,1,1)
plot (Irms);
title('Corriente RMS');
xlabel('muestras')
ylabel('A')
subplot (2,1,2)
plot(drms);
hold on
grid on
plot(locs,drms(locs),'rv','MarkerFaceColor','r');
%Dibuja los picos.
title('Eventos');
xlabel('muestras')
ylabel('A')
%figure(3)
hold off
```

## 7. PLIEGO DE CONDICIONES

En este apartado se adjuntan las hojas de características del chip ADE9153A y el diagrama de conexiones de la tarjeta que lo incorpora

### 7.1 CARACTERÍSTICAS DEL CHIP ADE9153

En este apartado se recogen las tablas y las gráficas relacionadas con los valores típicos y máximos del chip ADE9153

#### ABSOLUTE MAXIMUM RATINGS

T<sub>A</sub> = 25°C, unless otherwise noted.

Table 4.

Parameter	Rating
VDD to AGND/DGND	-0.3 V to +3.96 V
Analog Input Voltage to AGND/DGND, IAP, IAN, IBP, IBN, VP, VN <sup>1</sup>	-0.75 V to +2.2 V
Reference Input Voltage to AGND/DGND	-0.3 V to +2.2 V
Digital Input Voltage to AGND/DGND	-0.3 V to +3.96 V
Digital Output Voltage to AGND/DGND	-0.3 V to +3.96 V
Operating Temperature	
Industrial Temperature Range	-40°C to +85°C
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering, 10 sec) <sup>2</sup>	260°C
Electrostatic Discharge (ESD)	
Human Body Model (HBM)	4 kV
Machine Model (MM)	200 V
Field Induced Charged Device Model (FICDM)	1.25 kV

<sup>1</sup> The rating of -0.75 V on the analog input pins is limited by protection diodes inside the ADE9153A. These pins were tested with 7.5 mA going to the pin to simulate a 30x overcurrent condition on the channel, based on the test circuit antialiasing resistor of 150 Ω.

<sup>2</sup> Analog Devices, Inc., recommends that reflow profiles used in soldering RoHS-compliant devices conform to J-STD-020D.1 from JEDEC. Refer to JEDEC for the latest revision of this standard.

Stresses at or above those listed under Absolute Maximum Ratings may cause permanent damage to the product. This is a stress rating only; functional operation of the product at these or any other conditions above those indicated in the operational section of this specification is not implied. Operation beyond the maximum operating conditions for extended periods may affect product reliability.

#### THERMAL RESISTANCE

Thermal performance is directly linked to printed circuit board (PCB) design and operating environment. Careful attention to PCB thermal design is required.

θ<sub>JA</sub> and θ<sub>JC</sub> are specified for the worst-case conditions, that is, a device soldered in a circuit board for surface-mount packages.

Table 5. Thermal Resistance

Package Type	θ <sub>JA</sub> <sup>1</sup>	θ <sub>JC</sub> <sup>2</sup>	Unit
CP-32-12 <sup>3</sup>	27.83	2.10	°C/W

<sup>1</sup> The θ<sub>JA</sub> measurement uses a 252P JEDEC test board.

<sup>2</sup> The θ<sub>JC</sub> measurement uses a 150P JEDEC test board.

<sup>3</sup> All thermal measurements comply with JESD51.

#### ESD CAUTION



#### ESD (electrostatic discharge) sensitive device.

Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

## TYPICAL PERFORMANCE CHARACTERISTICS

### ENERGY LINEARITY OVER SUPPLY AND TEMPERATURE

Energy characteristics obtained from a 50% of full scale, sinusoidal, 50 Hz voltage signal; the sinusoidal, 50 Hz, swept amplitude current signal is from 100% of full scale to 0.01% of full scale.

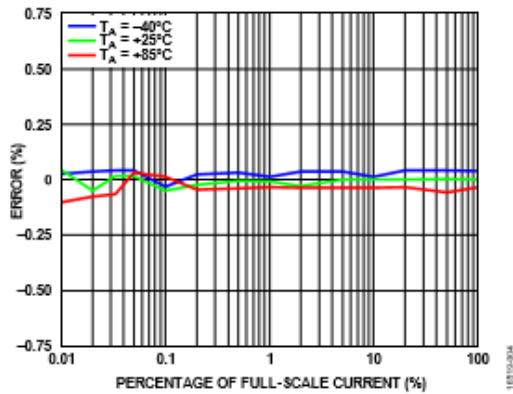


Figure 4. Total Active Energy Error as a Percentage of Full-Scale Current over Temperature, Power Factor = 1, Current Channel A (AI) PGA Gain = 16x

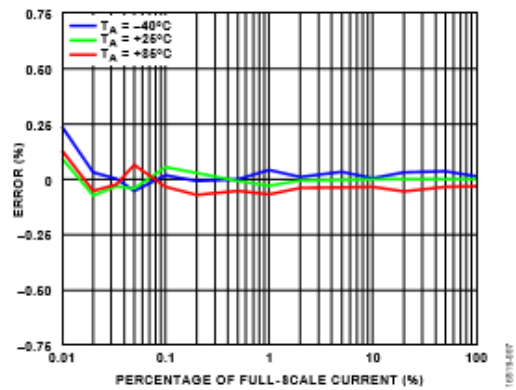


Figure 7. Fundamental Reactive Energy Error as a Percentage of Full-Scale Current over Temperature, Power Factor = 0, AI PGA Gain = 38.4x

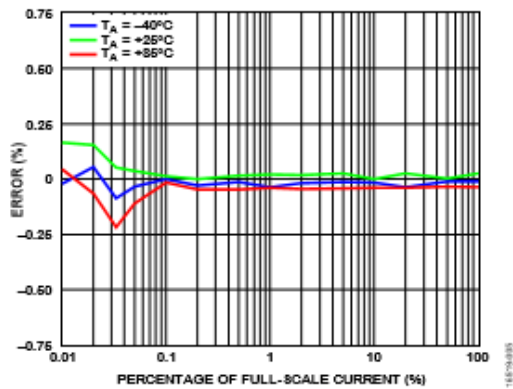


Figure 5. Total Active Energy Error as a Percentage of Full-Scale Current over Temperature, Power Factor = 1, AI PGA Gain = 38.4x

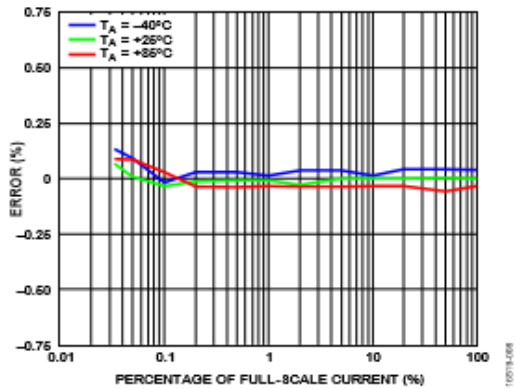


Figure 8. Total Apparent Energy Error as a Percentage of Full-Scale Current over Temperature, Power Factor = 1, AI PGA Gain = 16x

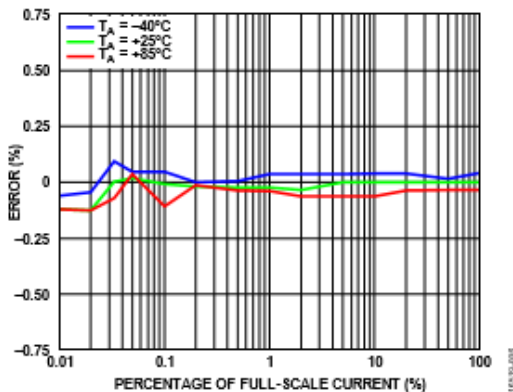


Figure 6. Fundamental Reactive Energy Error as a Percentage of Full-Scale Current over Temperature, Power Factor = 0, AI PGA Gain = 16x

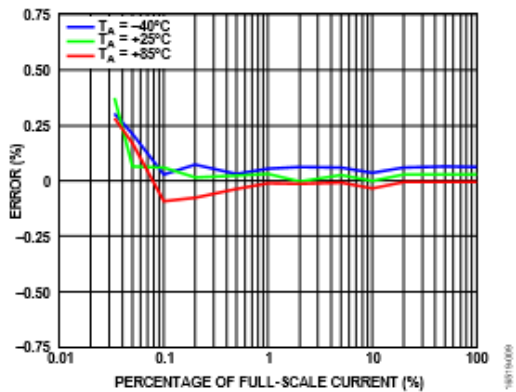


Figure 9. Total Apparent Energy Error as a Percentage of Full-Scale Current over Temperature, Power Factor = 1, AI PGA Gain = 38.4x

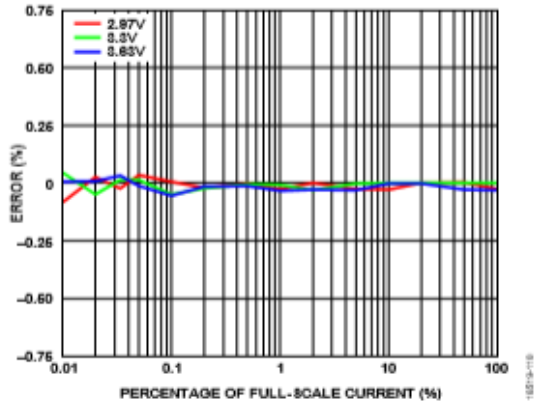


Figure 10. Total Active Energy Error as a Percentage of Full-Scale Current over Supply Voltage, Power Factor = 1,  $T_A = 25^\circ\text{C}$ , AI PGA Gain = 16x

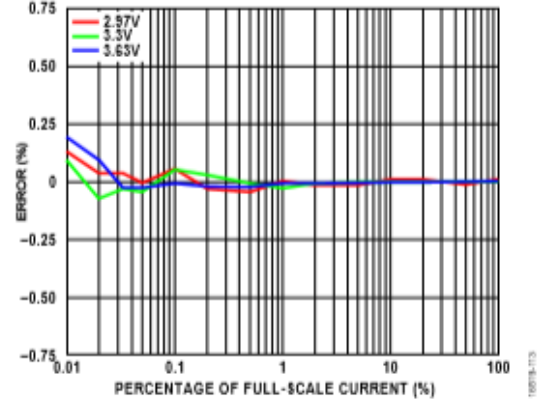


Figure 13. Fundamental Reactive Energy Error as a Percentage of Full-Scale Current over Supply Voltage, Power Factor = 0,  $T_A = 25^\circ\text{C}$ , AI PGA Gain = 38.4x

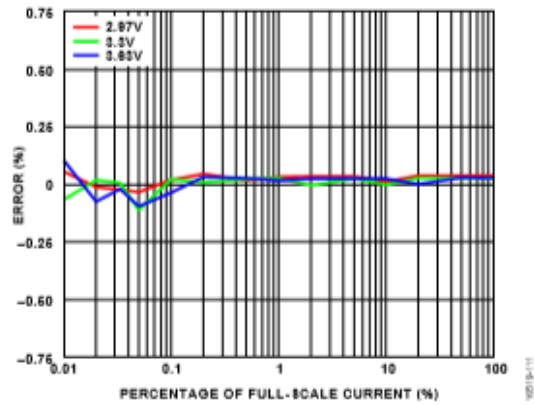


Figure 11. Total Active Energy Error as a Percentage of Full-Scale Current over Supply Voltage, Power Factor = 1,  $T_A = 25^\circ\text{C}$ , AI PGA Gain = 38.4x

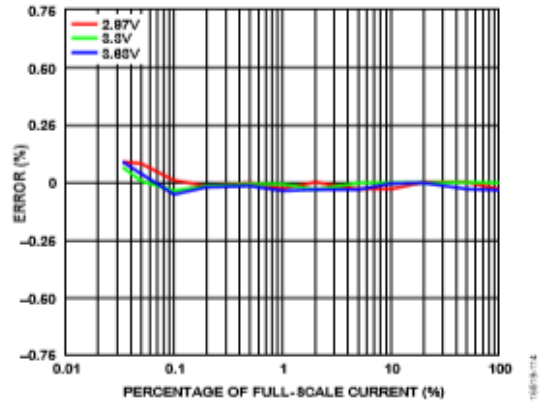


Figure 14. Total Apparent Energy Error as a Percentage of Full-Scale Current over Supply Voltage, Power Factor = 1,  $T_A = 25^\circ\text{C}$ , AI PGA Gain = 16x

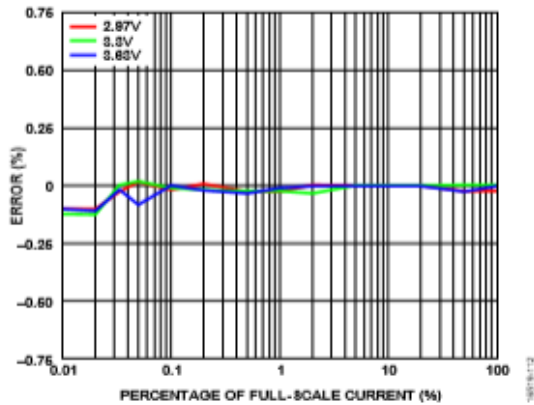


Figure 12. Fundamental Reactive Energy Error as a Percentage of Full-Scale Current over Supply Voltage, Power Factor = 0,  $T_A = 25^\circ\text{C}$ , AI PGA Gain = 16x

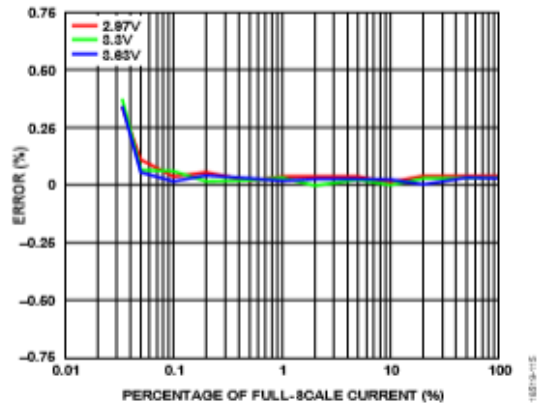


Figure 15. Total Apparent Energy Error as a Percentage of Full-Scale Current over Supply Voltage, Power Factor = 1,  $T_A = 25^\circ\text{C}$ , AI PGA Gain = 38.4x

**RMS LINEARITY OVER TEMPERATURE AND RMS ERROR OVER FREQUENCY**

RMS linearity obtained with a sinusoidal, 50 Hz current and voltage signals with a swept amplitude from 100% of full scale to 0.033% of full scale.

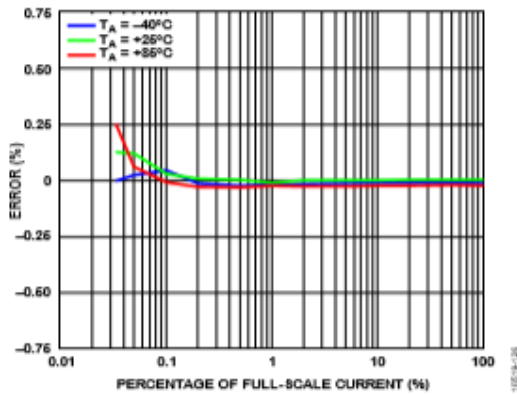


Figure 19. Current Channel A RMS Error as a Percentage of Full-Scale Current over Temperature, AI PGA Gain = 16x

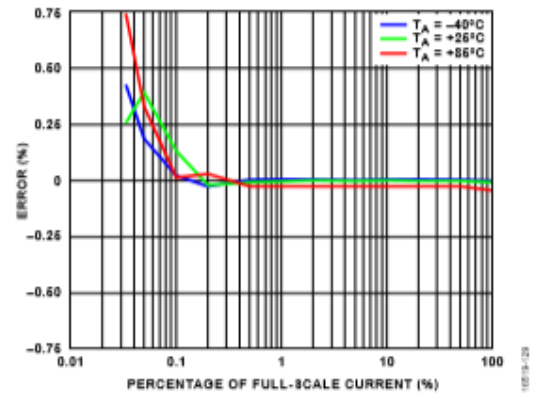


Figure 22. Voltage Channel RMS Error as a Percentage of Full-Scale Current over Temperature

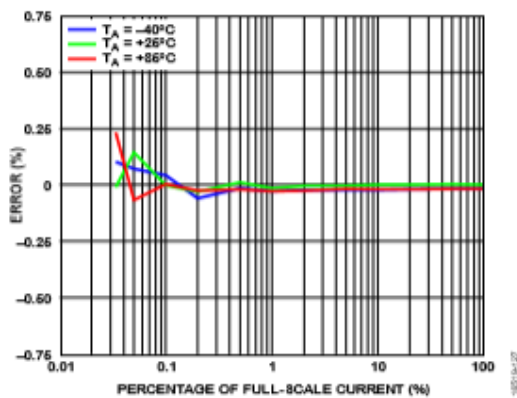


Figure 20. Current Channel A RMS Error as a Percentage of Full-Scale Current over Temperature, AI PGA Gain = 38.4x

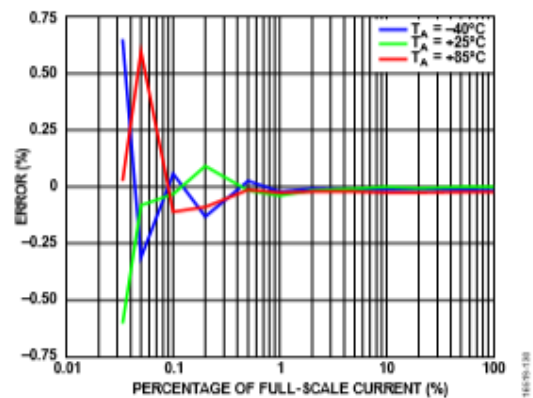


Figure 23. Current Channel A RMS Offset Error as a Percentage of Full-Scale Current over Temperature, AI PGA Gain = 16x

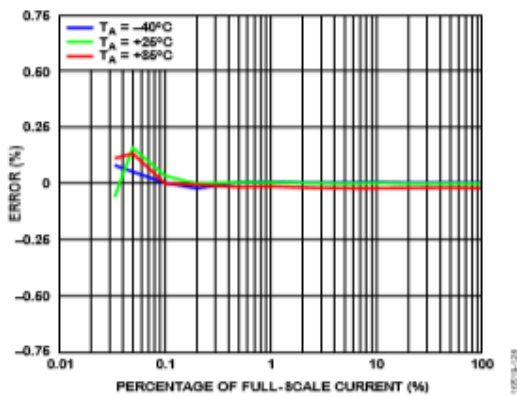


Figure 21. Current Channel B RMS Error as a Percentage of Full-Scale Current over Temperature

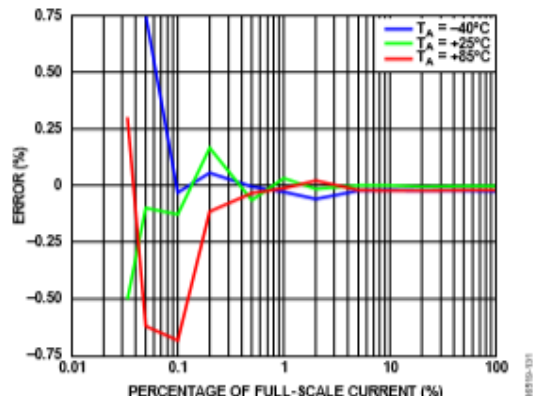


Figure 24. Current Channel A RMS Offset Error as a Percentage of Full-Scale Current over Temperature, AI PGA Gain = 38.4x

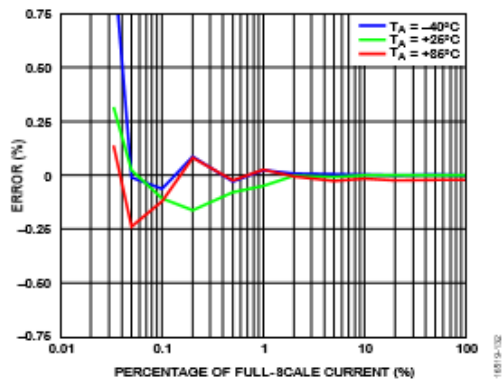


Figure 25. Current Channel B RMS Offset Error as a Percentage of Full-Scale Current over Temperature

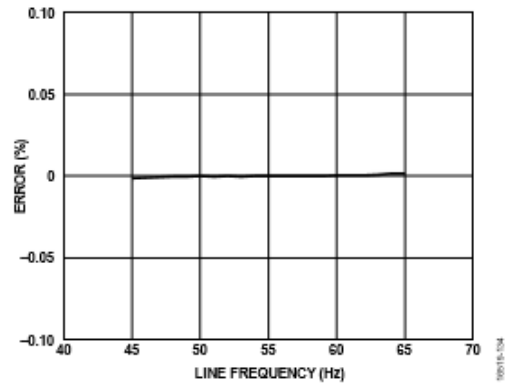


Figure 27. Current Channel A RMS Error vs. Line Frequency

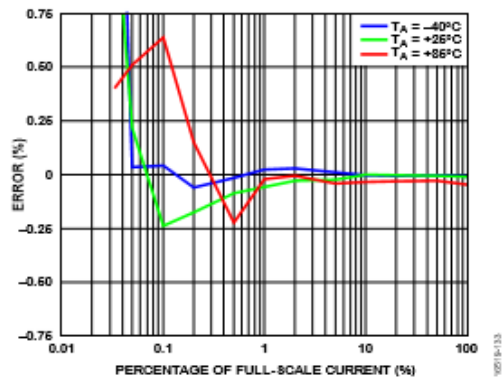


Figure 26. Voltage Channel RMS Offset Error as a Percentage of Full-Scale Current over Temperature

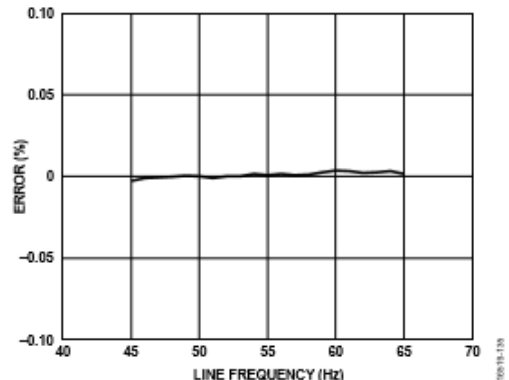


Figure 28. Current Channel B RMS Error vs. Line Frequency

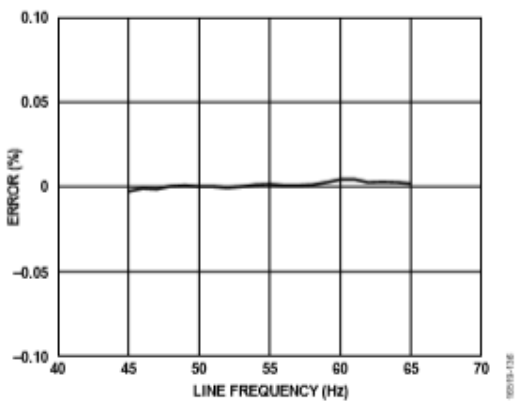


Figure 29. Voltage Channel RMS Error vs. Line Frequency

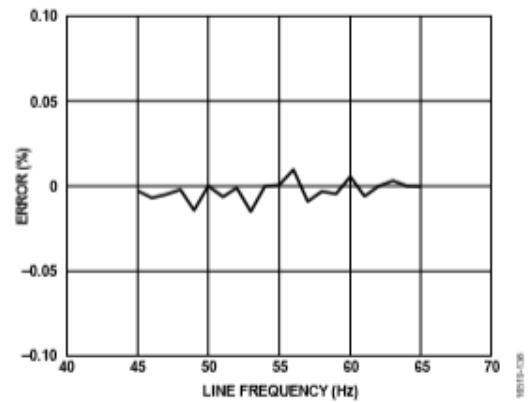


Figure 31. Current Channel B RMS Overcurrent Error vs. Line Frequency

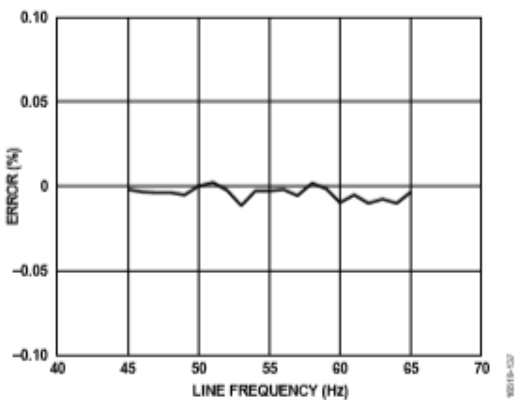


Figure 30. Current Channel A RMS Overcurrent Error vs. Line Frequency

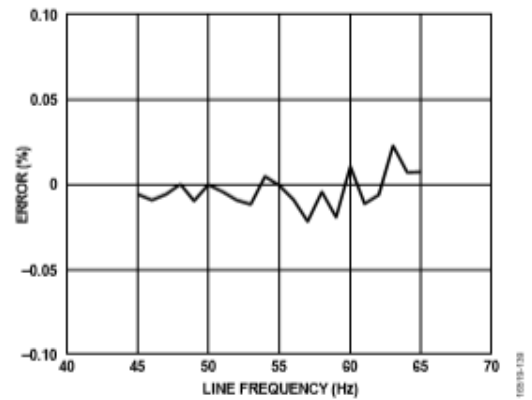


Figure 32. Voltage Channel RMS Overcurrent Error vs. Line Frequency



**SIGNAL-TO-NOISE RATIO (SNR) PERFORMANCE OVER DYNAMIC RANGE**

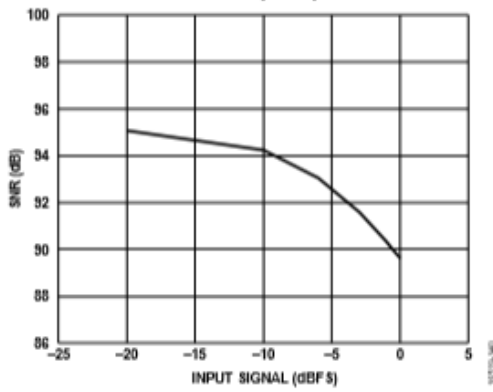


Figure 33. Current Channel A SNR with Respect to Full Scale, AI PGA Gain = 16x

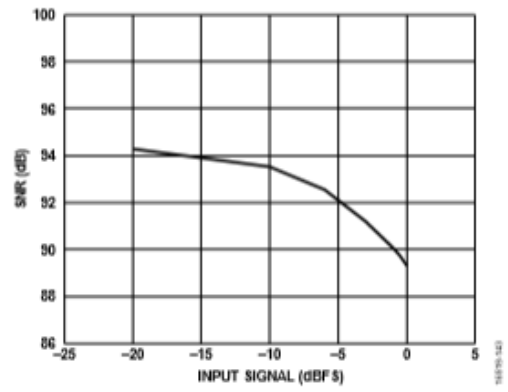


Figure 35. Current Channel B SNR with Respect to Full Scale

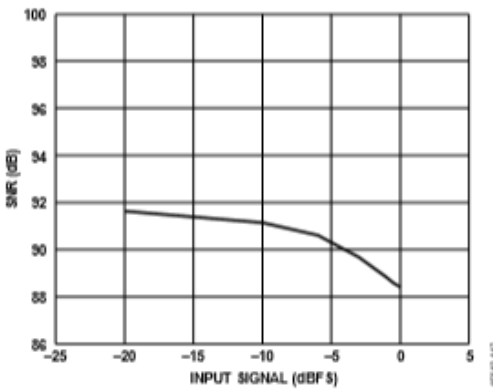


Figure 34. Current Channel A SNR with Respect to Full Scale, AI PGA Gain = 38.4x

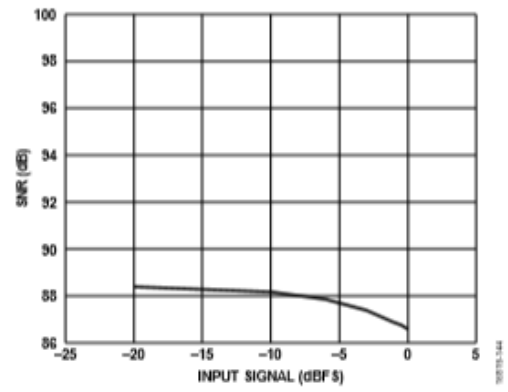


Figure 36. Voltage Channel SNR with Respect to Full Scale

## 7.2. DIAGRAMA DE CONEXIONES DE LA TARJETA

A continuación, se muestra el diagrama de conexiones de la tarjeta que incorpora el chip ADE9153A

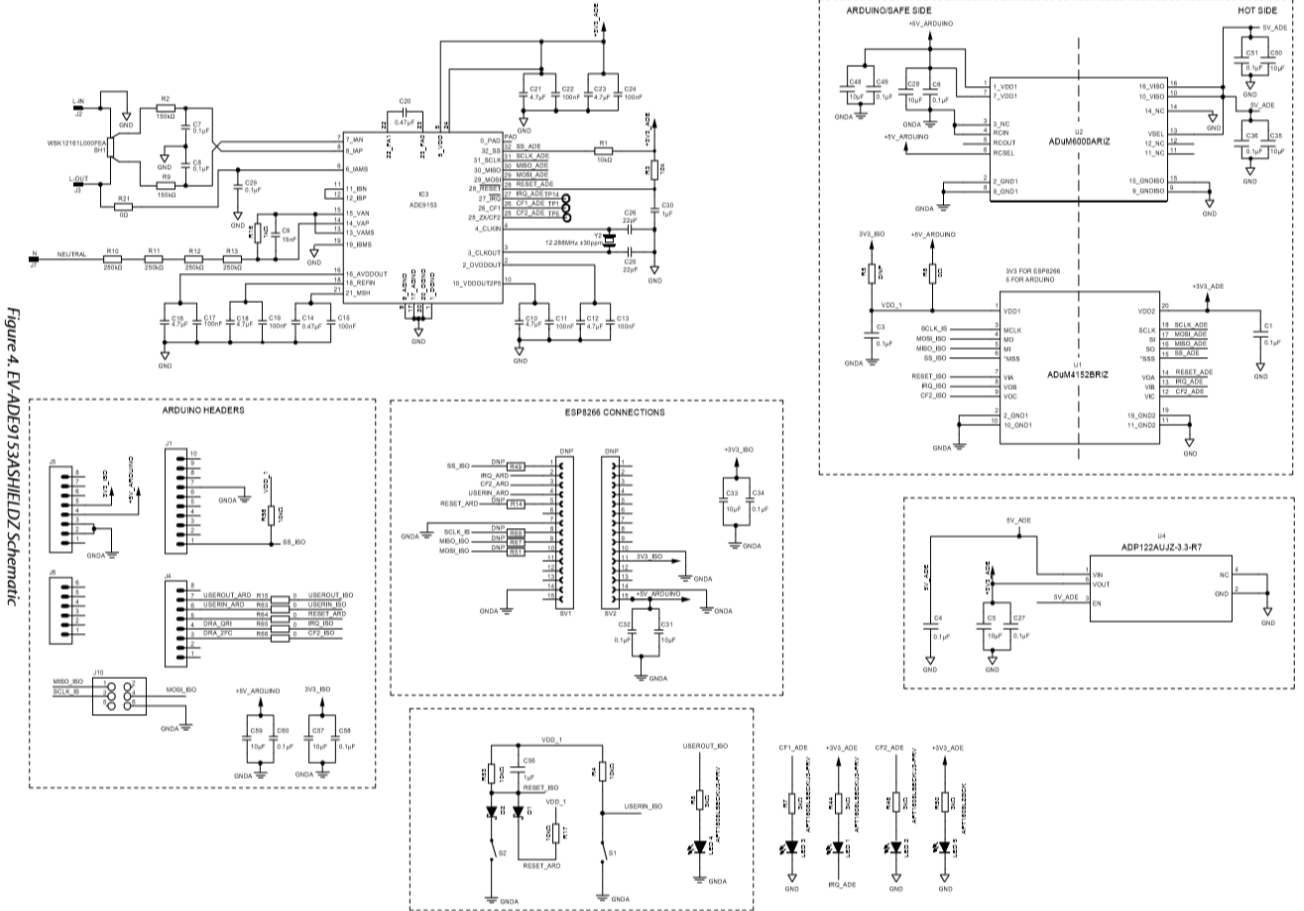


Figure 4. EV-ADE9153A SHIELD Schematic

## 8. PRESUPUESTO

En este apartado se detalla el presupuesto estimado para la realización de este proyecto.

En la tabla 5 se recoge el valor de los equipos empleados, calculados para una amortización por un periodo de uso de 3 años (al ser material informático) y un uso de un año [16].

DESCRIPCIÓN	CANTIDAD	PRECIO (€)	SUBTOTAL (€)
Ordenador	1	1.000,00 €	333,33 €
EV-ADE9153ASHIELDZ	1	120,00 €	40,00 €
Módulo de Arduino	1	90,00 €	30,00 €
Material para pruebas	1	150,00 €	50,00 €
Total			453,33 €

Tabla 5: Coste equipo empleado

En la tabla 6 se muestra el valor de la licencia del software empleado.

DESCRIPCIÓN	CANTIDAD	PRECIO (€)	SUBTOTAL
Licencia Matlab	1	119,00 €	119,00 €
Total			119,00 €

Tabla 6: Coste software empleado

En la tabla 7 aparece el coste de personal, calculado a partir de las horas efectivas dedicadas al proyecto, realización de pruebas y redacción.

DESCRIPCIÓN	CANTIDAD	PRECIO (€/h)	SUBTOTAL
Tareas de ingeniería	500	60,00 €	30.000,00 €
Total			30.000,00 €

Tabla 7: Coste de personal

El coste total de este proyecto, aplicando un I.V.A del 21 % es de **36.992,52 €**.

DESCRIPCIÓN	PRECIO (€)
Equipos	453,33 €
Software	119,00 €
Coste de personal	30.000€
I.V.A.	6420,19 €
Total	36.992,52 €

Tabla 8: Coste total del proyecto

## 9. MANUAL DE USUARIO

A continuación, se detalla un manual de usuario recogiendo el equipo hardware y el software necesario como el conexionado y los distintos pasos necesarios para llegar a la obtención de los resultados oportunos según las cargas que se conecten.

### 9.1 HARDWARE Y SOFTWARE NECESARIO

Para llevar a cabo el presente proyecto se necesita el siguiente material hardware:

1. Tarjeta de adquisición EV-ADE9153ASHIELDZ
2. Módulo de Arduino One.
3. Regleta para conectar las cargas y cable de alimentación para la toma de red.

En cuanto al software:

1. IDE de Arduino para cargar el programa
2. Matlab para cargar el algoritmo´
3. Programa hiperterminal Realterm para poder almacenar los datos en un archivo de texto.

### 9.2 MONTAJE HARDWARE

Para realizar el montaje hardware es necesario seguir los siguientes pasos:

1. Primero se conecta el cable de alimentación a la tarjeta de adquisición. Para ello se suelda el neutro y el negativo del cable al neutro de la tarjeta. La fase del cable se conecta a la entrada de la tarjeta.
2. Posteriormente se conecta la regleta, soldando el neutro con el neutro de la tarjeta y la fase con la entrada de carga de la tarjeta.

En la figura 32 se muestra el conexionado entre la tarjeta, la toma de alimentación y la carga.

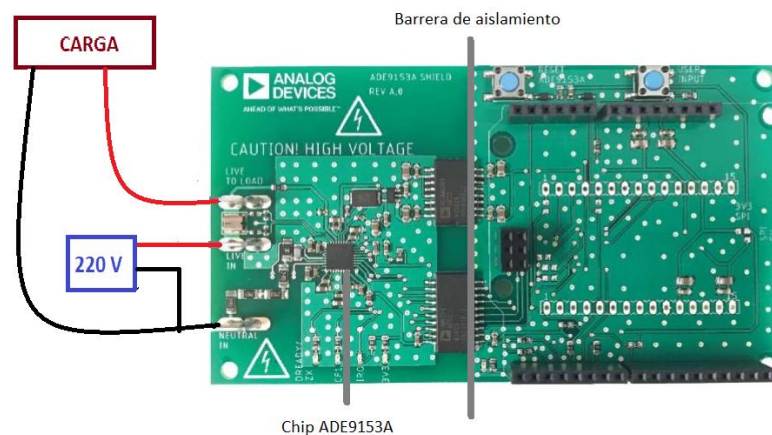


Figura 32: Conexión entre la tarjeta, la carga y la toma de alimentación

3. Por último, se conecta el módulo Arduino con la tarjeta.  
En la figura 33 se observa la disposición de la tarjeta junto con el Arduino

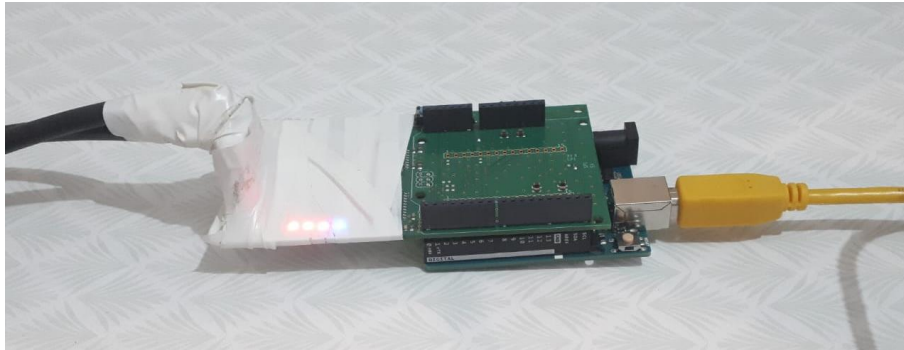


Figura 33: Tarjeta con módulo Arduino conectado.

Una de las partes de la tarjeta funciona a 220V por lo que es conveniente envolver dicha parte con cinta a modo de protección para el usuario, así como la conexión. También se debe gripar cada punto de soldadura con el fin de que tanto los cables como el pin de la tarjeta queden adheridos por completo.

### 9.3 SOFTWARE.

Primero se debe descargar el software proporcionado por el fabricante desde la página web de Analog Device e instalar el IDE de Arduino.

Posteriormente se debe crear el proyecto sobre Arduino, para ello se siguen los siguientes pasos:

1. Todos los proyectos de Arduino se deben localizar en una carpeta que se llame igual que el programa principal. Así, en el proyecto presente todos los archivos que incluye el .zip del fabricante se deben descomprimir dentro de una carpeta que se llame “*ADE9153AAPI\_Test*”. Si se intenta abrir el programa principal y no se localiza en esta carpeta, Arduino creará dicha carpeta y moverá el fichero a ésta. Si esto ocurre se deben trasladar tanto las librerías como el código fuente a la carpeta.

Este paso es muy importante a la hora de que el programa compile correctamente y el programa principal enlace con el código fuente.

2. Posteriormente se deben cargar las librerías del proyecto en Arduino, para ello:

- En el IDE de Arduino:

*Programa > Incluir librería > Añadir biblioteca .ZIP.*

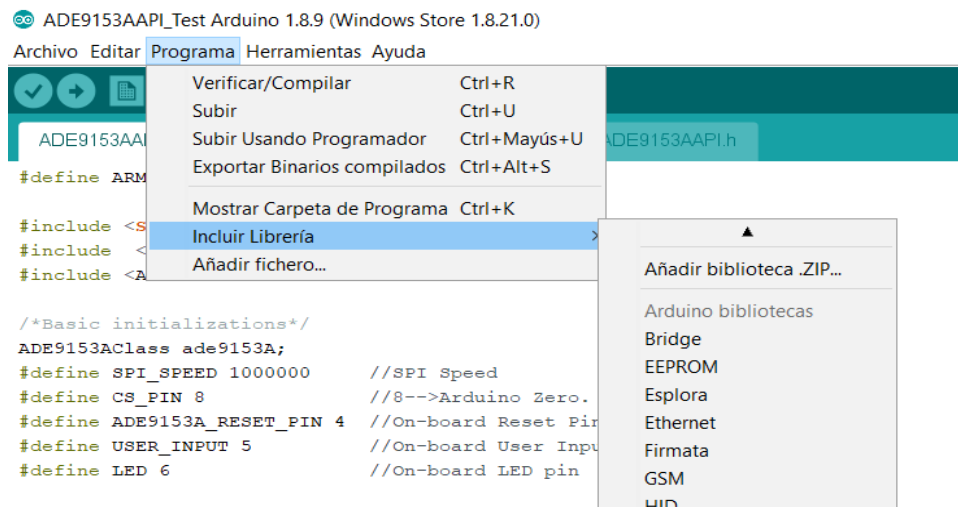


Figura 34: Incluir librería en el proyecto.

- Se abre un cuadro de dialogo para seleccionar el archivo .ZIP deseado, tal y como el que se muestra en la figura 35. Se debe seleccionar el .ZIP que proporciona el fabricante “ade9153a\_arduino\_software.zip”.

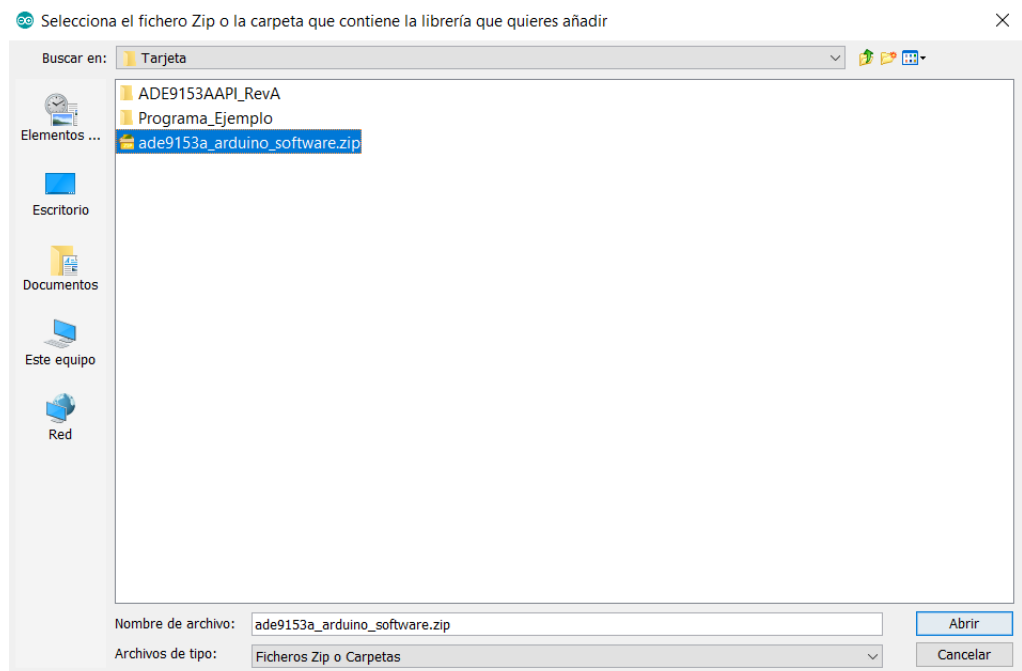
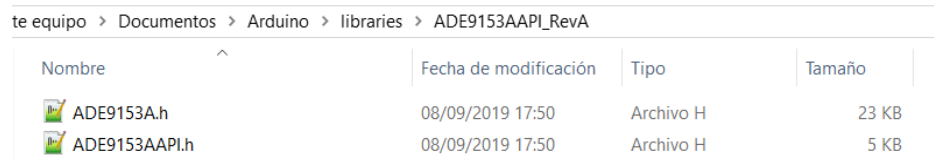


Figura 35: Cuadro de dialogo para incluir las librerías al proyecto

- Haciendo esto el IDE instala todos los ficheros que se encuentren en el archivo, luego en este caso instala tanto las librerías como el código fuente. Este hecho produce un error de compilación ya que el IDE reconoce que las constantes y variables globales están definidas varias veces dentro del proyecto. Para solventarlo hay que eliminar el archivo del código fuente de la propia carpeta en la que Arduino guarda los archivos de las librerías instaladas, dejando los otros dos archivos tal y como se muestra en la figura 36. Las librerías instaladas suelen estar en la siguiente ruta:  
*Arduino > libraries > ADE9153AAPI\_RevA*



Nombre	Fecha de modificación	Tipo	Tamaño
ADE9153A.h	08/09/2019 17:50	Archivo H	23 KB
ADE9153AAPI.h	08/09/2019 17:50	Archivo H	5 KB

Figura 36: Librerías a incluir

3. Una vez que se ha creado el proyecto, hay que modificar el código para solventar los errores y aplicar las soluciones que se han comentado en el apartado “5.7.1 DATOS” para obtener datos coherentes.
4. Con los errores resueltos, se compila el proyecto, para ello hay que clicar en el botón de “Verificar” que incorpora el IDE en su barra de herramientas, o seleccionar la opción “Verificar/Compilar” en la pestaña de “Programa”. Si el código o el proyecto tiene errores estos se reflejarán en la parte de compilación a modo de mensaje.
5. Si no se detectan errores tras la compilación se puede cargar el proyecto en el módulo de Arduino. Para ello primero se conecta el módulo de Arduino a un puerto del ordenador y se pulsa el botón de “Subir” que se encuentra también en la barra de herramientas. También se puede seleccionar la opción “Subir” en la pestaña de “Programa”,  
No es necesario configurar el puerto serie en el IDE ya que éste lo detecta en el momento que se conecta el módulo.
6. Por último, para poder visualizar los datos, se abre el monitor serie que incorpora el IDE pulsando el botón situado en la parte superior derecha de la interfaz con forma de lupa.

## 9.4 CAPTURA DE DATOS

Como se ha especificado en apartados anteriores es necesario almacenar los datos en un fichero de texto para que puedan ser procesados en Matlab, mediante el programa Realterm.

Para configurar éste hay que seguir los siguientes pasos:

1. Hay que configurar el puerto y la velocidad de transmisión del puerto serie a 115200 baudios en la pestaña “Port”. En la figura 37 se muestra la configuración.

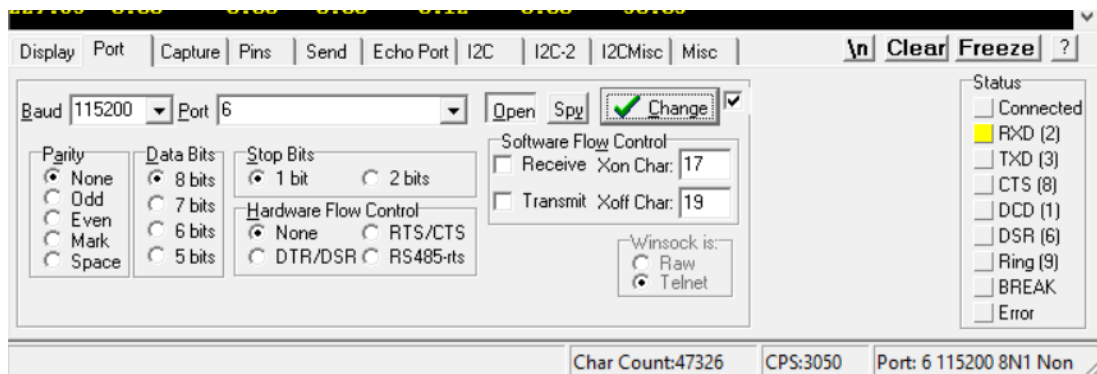


Figura 37: Configuración de la UART en el programa Realterm

2. También hay que seleccionar el formato de texto en el que se recibe los datos, así en la pestaña “Display” se ha de seleccionar el formato *Ansi*.
3. Una vez que se han configurado los distintos parámetros se puede proceder a mandar los datos a un fichero de texto, para ello en la pestaña “Capture” se han de realizar los siguientes pasos:
  - Seleccionar el fichero que se va a escribir: Este fichero se tiene que ubicar en un directorio distinto al C, como por ejemplo el directorio de datos. Teniendo esto en cuenta, en la pestaña “file” se selecciona el desplegable en el que se puede elegir la ruta donde guardar los datos y el nombre del fichero.
  - Cuando se quiera proceder a almacenar los datos en el fichero se pulsa el botón “Start Overwrite” y la pestaña se pondrá en rojo indicando que los datos se están escribiendo en un fichero, cuando se quiera detener se pulsa “Stop Capture”.
  - Si se quiere continuar escribiendo en el mismo fichero sin sobrescribir los datos anteriores se debe pulsar “Start Append”, si por el contrario se quieren borrar los datos anteriores se pulsa de nuevo “Start Overwrite”.
  - También se puede seleccionar el formato en la ventana “TimeStamp” y exportar los datos al formato empleado por Matlab, seleccionando su opción.



En la figura 38 se pueden observar todos los botones de la pestaña “Capture”.

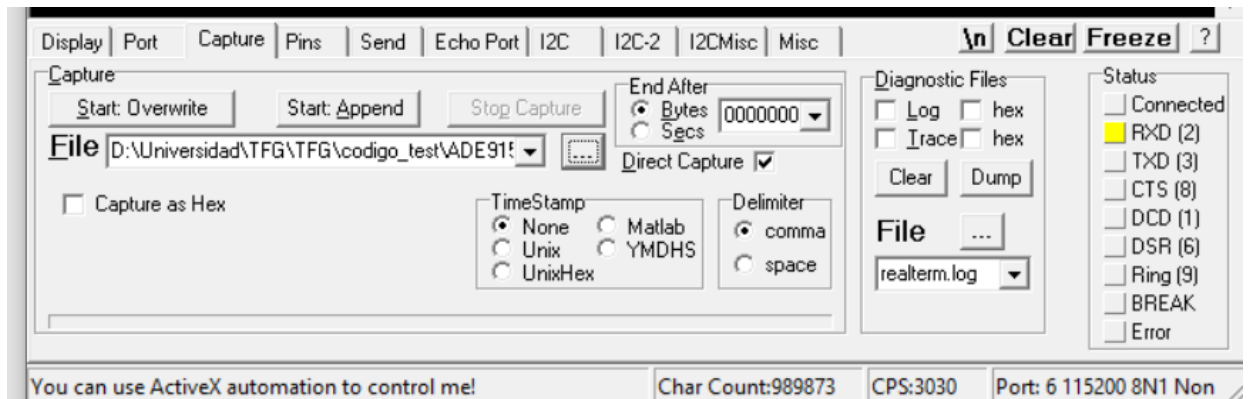


Figura 38: Pestaña capture del programa Realterm

Una vez que ya se tienen los datos en un fichero de texto o en un fichero .m se cargan en Matlab y se aplica el algoritmo explicado en la memoria para obtener la detección de eventos de las distintas cargas.

## 10. BIBLIOGRAFIA

- [1] Emma Burés, «¿Cómo funciona la desagregación de consumos? Mirubee,» Marzo 2018. [En línea]. Available: <https://mirubee.zendesk.com/hc/es/articles/115002218652--C%C3%B3mo-funciona-la-desagregaci%C3%B3n-de-consumos->.
- [2] J. M. A. Orzáez, *Non-Intrusive Load Monitoring techniques for Activity of Daily Living recognition*, Alcalá de Henares, 2016.
- [3] l. e. l. Wikipedia, «Nonintrusive Load monitoring,» [En línea]. Available: [https://en.wikipedia.org/wiki/Nonintrusive\\_load\\_monitoring](https://en.wikipedia.org/wiki/Nonintrusive_load_monitoring).
- [4] O. Parson, «Disaggregated Homes,» 25 Marzo 2015. [En línea]. Available: <http://blog.oliverparson.co.uk/2015/03/overview-of-nilm-field.html>.
- [5] Sara Ledo, «El Periódico,» 21 Octubre 2018. [En línea]. Available: <https://www.elperiodico.com/es/economia/20181021/espana-mide-luz-contadores-inteligentes-7098751>.
- [6] Iberdrola S.A., «Proyecto STAR,» [En línea]. Available: <https://www.iberdrola.com/conocenos/lineas-negocio/proyectos-emblematicos/proyecto-star>.
- [7] SMARTGRIDSINFO, «PRIME Alliance,» [En línea]. Available: <https://www.smartgridsinfo.es/prime-alliance>.
- [8] T. García, «La gestión de la red que lleva la electricidad a los hogares,» 4 Junio 2014. [En línea]. Available: <https://www.blog.iberdrola.com/2014/06/04/la-gestion-de-la-red-que-lleva-la-electricidad-a-los-hogares/>.
- [9] Mirubee, «Manual Mirubox MONO y TRI,» [En línea]. Available: <https://mirubee.zendesk.com/hc/es/categories/115000213032-%C3%81REA-T%C3%89CNICA>.
- [10] Mirubee, «Virtual Submetering,» [En línea]. Available: <https://mirubee.com/es/info/12-virtual-submetering>.
- [11] Wibeee, «Wibeee,» [En línea]. Available: <http://wibeee.circuitor.com/#>.
- [12] T. J. F. a. A. N. Hassan, *An empirical investigation of v-i trajectory based load signatures for non-intrusive load monitoring.*, 2014.
- [13] Analog Devices, *ADE9153A Technical Reference Manual UG-1247*, 2018.
- [14] Analog Devices, *Energy Metering IC with Autocalibration. Data Sheet*, 2018.
- [15] Analog Devices, *EV-ADE9153ASHIELDZ User Guide*.

- [16] Agencia Tributaria, [En línea]. Available: [agenciatributaria.es/AEAT.internet/Inicio/\\_Segmentos\\_/Empresas\\_y\\_profesionales/Empresas/Impuesto\\_sobre\\_Sociedades/Periodos\\_impositivos\\_a\\_partir\\_de\\_1\\_1\\_2015/Base\\_imponible/Amortizacion/Tabla\\_de\\_coeficientes\\_de\\_amortizacion\\_lineal\\_.shtml](http://agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml).
- [17] A. Heredia, *AN-1571 APPLICATION NOTE*.
- [18] MathWorks, «Documentación,» [En línea]. Available: <https://es.mathworks.com/help/matlab/ref/diff.html>.
- [19] MathWorks, «Documentación,» [En línea]. Available: <https://es.mathworks.com/help/signal/ref/findpeaks.html>.
- [20] «Realterm,» [En línea]. Available: <https://sourceforge.net/projects/realterm/>.
- [21] A. Sánchez, «Proconingenieros,» [En línea]. Available: <http://proconingenieros.com/consumo-electrico-de-una-lavadora/>.

## APENDICE 1

En este apéndice se recogen los informes generados por el Mirubox mono conectado para una vivienda de dos personas desde el mes de junio del 2019 hasta el mes de agosto del mismo año.

Si se observan se pueden obtener ciertas conclusiones en cuanto a los hábitos de las personas que habitan dicha vivienda, como por ejemplo los horarios que tienen ya que el consumo está concentrado prácticamente a partir de las 18:00 más o menos. A su vez también se puede observar cómo en los meses de junio y julio el consumo se eleva debido a que son meses en los que hace bastante calor o como durante el mes de agosto que coincide con el periodo vacacional el consumo disminuye y los hábitos de consumo se modifican.

- **JUNIO 2019**



### Informe de "Mi casa"

Junio 2019

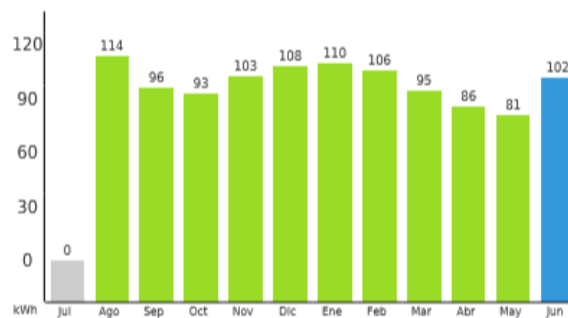
#### EVOLUCIÓN DE TU CONSUMO

Último mes:  
102 kWh  
52,22€\*



+26 %

\*Tú consumo ha sido +26 % mayor respecto al mes anterior.



\*Sólo el coste de la energía consumida, sin contar términos fijos que pudiera haber (por potencia contratada por ejemplo), alquiler de equipos, etc.

#### USOS MÁS RELEVANTES



STANDBY \*  
28 kWh  
27.77%



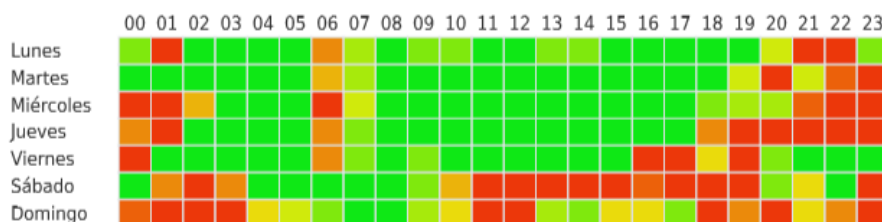
FRIGORÍFICO  
19 kWh  
18.62%



LAVADORA  
2 kWh  
2.42%

standby \*Dispositivos permanentemente conectados aunque no estén encendidos.

#### DISTRIBUCIÓN SEMANAL



Bajo consumo: <0.09 kWh      0.14 kWh      Alto consumo: >0.19 kWh

• JULIO 2019



Informe de "Mi casa"  
Julio 2019

EVOLUCIÓN DE TU CONSUMO

Último mes:  
133 kWh  
71,8€\*  
  
+30 %

\*Tú consumo ha sido +30 % mayor respecto al mes anterior.



\*Sólo el coste de la energía consumida, sin contar términos fijos que pudiera haber (por potencia contratada por ejemplo), alquiler de equipos, etc.

USOS MÁS RELEVANTES



STANDBY \*  
31 kWh  
23.35%



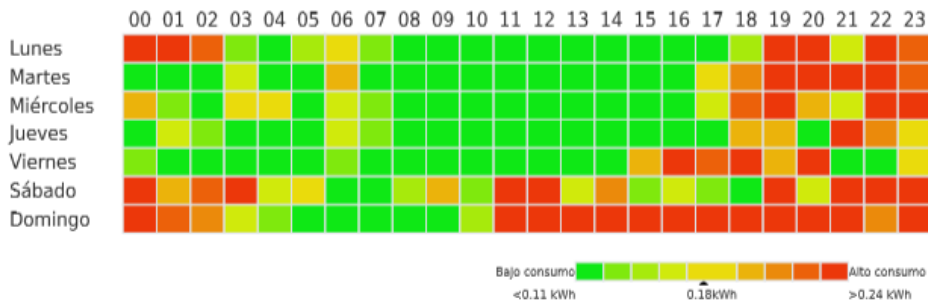
FRIGORÍFICO  
25 kWh  
18.49%



LAVADORA  
2 kWh  
1.33%

standby \*Dispositivos permanentemente conectados aunque no estén encendidos.

DISTRIBUCIÓN SEMANAL



• **AGOSTO 2019**

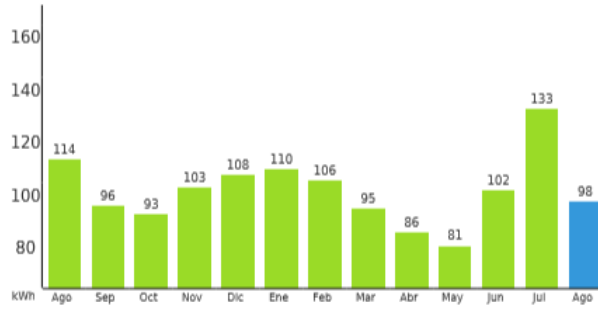


Informe de "Mi casa"  
Agosto 2019

EVOLUCIÓN DE TU CONSUMO

Último mes:  
98 kWh  
50,83€\*  
↓  
-26 %

\*Tú consumo ha sido -26 % menor respecto al mes anterior.



\*Sólo el coste de la energía consumida, sin contar términos fijos que pudiera haber (por potencia contratada por ejemplo), alquiler de equipos, etc.

USOS MÁS RELEVANTES



FRIGORÍFICO  
22 kWh  
22.64%



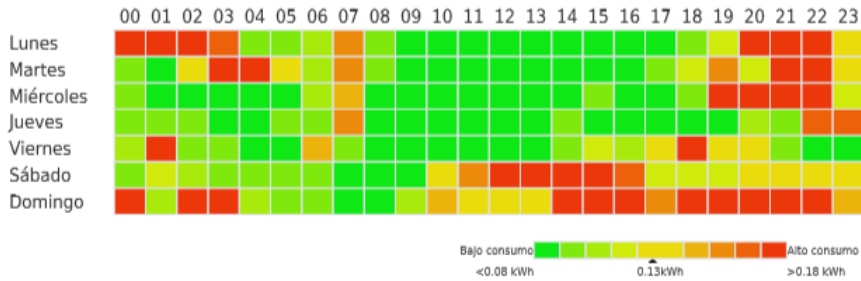
STANDBY \*  
15 kWh  
15.3%



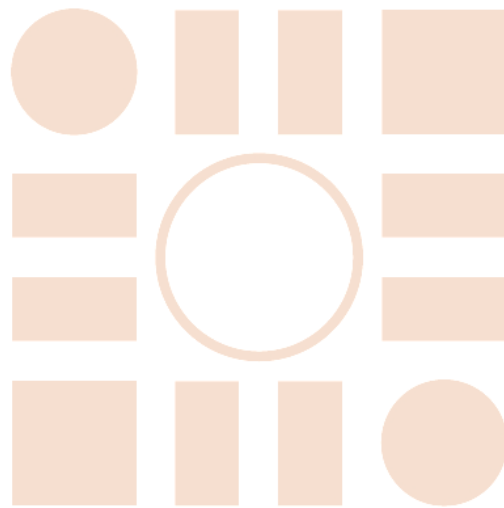
LAVADORA  
2 kWh  
2.51%

**standby** \*Dispositivos permanentemente conectados aunque no estén encendidos.

DISTRIBUCIÓN SEMANAL



Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá