



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Instituto  
Ingeniería  
Energética



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA

**TRABAJO FIN DE MÁSTER**  
**TECNOLOGÍA ENERGÉTICA PARA DESARROLLO SOSTENIBLE**

---

**DISEÑO E IMPLEMENTACIÓN DE UN  
EQUIPO CAPTURA Y REPORTE  
INALÁMBRICO DE DATOS DE  
CONSUMO ELÉCTRICO EN VIVIENDA  
UNIFAMILIAR PARA SU USO EN  
SISTEMAS DE GESTIÓN ENERGÉTICO  
DE EDIFICIOS**

---

**AUTOR: DASI BASTANTE, MIGUEL**

**TUTOR: PUCHE PANADERO, RUBÉN**

**COTUTOR: PINEDA SÁNCHEZ, MANUEL**

**Curso Académico: 2017-18**

**“Fecha 08/2018”**



## Contenido

1.	Introducción y objetivos.....	1
2.	Fundamentos teóricos.....	3
2.1.	Amplificador operacional.....	5
2.1.1.	Lazo abierto.....	6
2.1.2.	Lazo cerrado.....	7
2.2.	Tipos de convertidores ADC.....	10
2.2.1.	Entrada diferencial y entrada referida a masa.....	10
2.2.2.	Salidas en serie y en paralelo.....	12
2.2.3.	Codificación del dato de salida.....	13
2.2.4.	Topología de conversión.....	15
2.3.	Conexión de convertidores ADC a sistemas digitales.....	18
2.3.1.	Convertidores con salida en paralelo.....	18
2.3.2.	Convertidores con salida serie.....	19
2.3.3.	Convertidores A/D empotrados.....	23
3.	Análisis de alternativas y descripción de la solución adoptada.....	24
3.1.	Selección de sensores.....	24
3.2.	Tensión de referencia.....	26
4.	Diseño de los sensores de tensión y corriente.....	30
4.1.	Características de la red del suministro doméstico.....	30
4.2.	Sensores.....	30
4.2.1.	Diseño de los sensores.....	31
5.	Captura digital.....	45
5.1.	Microcontrolador.....	45
5.2.	Convertidor ADC.....	46
5.2.1.	MCP3008.....	46
5.2.2.	MCP3304.....	49
5.3.	Comunicación convertidor ADC - microcontrolador.....	52
5.3.1.	Comunicación SPI.....	52
6.	Determinación del precio de la energía.....	66
7.	Programación.....	78
7.1.	Arduino.....	78
7.1.1.	Librerías utilizadas y declaración de variables globales.....	78
7.1.2.	Función setup.....	79
7.1.3.	Función loop.....	80

7.2. Android Studio .....	93
7.2.1. Manual de usuario.....	99
8. Conclusiones.....	105
Planteamiento del presupuesto.....	106
Mano de obra.....	106
Materiales .....	107
Unidades de obra. Precios descompuestos .....	108
Unidades de obra. Precios unitarios .....	112
Presupuesto .....	112
Bibliografía .....	114
Anexos.....	116

## 1. Introducción y objetivos

A lo largo del presente trabajo se aborda el diseño e implementación de un equipo de captura y reporte inalámbrico de datos de consumo eléctrico en vivienda unifamiliar para su uso en sistemas de gestión energético de edificios.

La motivación de este proyecto deriva de la necesidad de monitorizar el consumo energético, ya sea de una vivienda unifamiliar como es en este caso, o de cualquier edificio o instalación eléctrica, para tener datos fiables en tiempo real que nos permita hacer un uso más eficiente de la energía. De esta forma, con la capacidad de monitorizar el consumo se puede identificar qué cargas están en funcionamiento, o incluso en el caso de implantar medidas de eficiencia energética poder cuantificar a partir de los históricos el ahorro energético y económico de las mismas.

Así pues, para el diseño del equipo en este trabajo se aborda el caso de una vivienda unifamiliar, pero como se discutirá más adelante, estas mismas pautas de diseño se podrán utilizar con el fin de instalar el equipo para monitorizar instalaciones más grandes o líneas individuales dentro de una instalación para poder monitorizar datos de consumo energético. Como se acaba de destacar, el consumo energético resulta ser una magnitud muy importante debido al coste que tiene asociado, pero también, además, se monitorizarán otras magnitudes como la tensión, que resulta interesante desde el punto de vista de la calidad de suministro de la red.

De este modo, una vez discutida la importancia de monitorizar el consumo eléctrico, en la actualidad podemos ver cómo diversos fabricantes han abordado el problema planteando soluciones. La solución más extendida y económica, consiste en un dispositivo que se conecta entre el enchufe y la carga que se tenga. Así pues, en esta área, podemos destacar fabricantes como smappee<sup>(1)</sup>, y en la siguiente imagen podemos ver el diseño del dispositivo:



Figura 1. Dispositivo de medición del consumo eléctrico (Smappee)

El problema que plantea este tipo de dispositivos es que únicamente muestran los valores instantáneos, sin posibilidad de consultar el histórico, además de que resulta inviable poder tener el consumo total en una vivienda puesto que por unidad estos dispositivos tienen un precio que ronda los 10€, y teniendo en cuenta que el gasto promedio de la factura eléctrica es de 82,5€, según la Organización del consumidor<sup>(2)</sup>, tendríamos que la inversión tardaría en amortizarse, debido a los ahorros potenciales derivados del control del consumo (ejemplo apagar luces, desconectar aparatos, etc.). Asimismo, el hecho de no contar con históricos dificulta el hecho de que el usuario pueda identificar consumos anómalos. Ante esta situación existe soluciones más avanzadas, que si son capaces de monitorizar el consumo del cuadro eléctrico y poder guardar los datos en un histórico<sup>(3)</sup>, sin embargo, el coste de estos otros dispositivos es mayor, desde unos 180€ el modelo más económico.

Todo esto provoca que el hecho de dar la posibilidad al usuario de monitorizar su consumo haga que no sea atractivo desde un punto de vista económico. Sin embargo, se podría diseñar un dispositivo más económico, basado en la medición de tensión y corriente mediante sensores de bajo coste, con la posibilidad de poder mostrar históricos para facilitar al usuario el seguimiento de su consumo. Será este el objetivo que se seguirá en los próximos capítulos.

## 2. Fundamentos teóricos.

Para la comprensión del presente proyecto, se hace necesario describir una serie de fundamentos teóricos que ayudarán al lector a la comprensión del documento en los capítulos posteriores donde se aborda la solución técnica del problema planteado en el capítulo anterior.

Puesto que resulta complejo abordar los diferentes aspectos generales de la instrumentación electrónica en un capítulo, se describirán los aspectos únicamente útiles para la comprensión de este proyecto.

Así pues, como se describe en los siguientes capítulos, cualitativamente la solución adoptada consiste en sensores basados en resistencias funcionando a modo de divisor de tensión, cuya señal de tensión será leída por un convertidor analógico a digital (en adelante ADC), que transmitirá dicho valor en digital a un microcontrolador para leer dicha información y tratarla para que finalmente se puedan presentar los datos al usuario haciendo uso de un dispositivo Android. El trasiego de información entre el microcontrolador y el dispositivo Android, será posible enviando y recibiendo datos a la nube, mediante protocolo MQTT.

Como primera aproximación, se podría resumir esta solución gráficamente como se indica en la siguiente imagen:

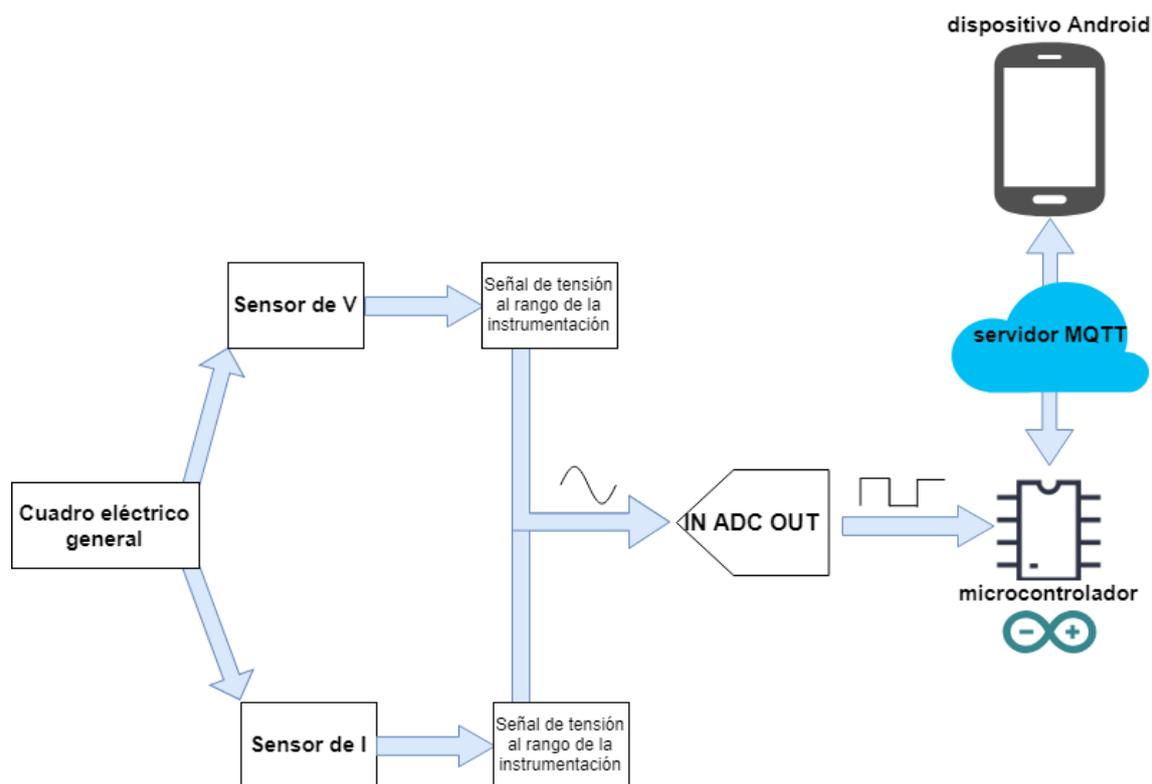


Figura 2. Solución adoptada

Donde para los sensores de tensión y corriente será necesario fijar un valor de tensión de referencia. El motivo de fijar esta tensión es debido a que las señales de entrada al convertidor ADC deben ser positivas (mayor que cero), mientras que la onda de tensión y corriente tiene valores positivos para un semi-periodo de onda y negativos en el otro semi-periodo. Es por esto por lo que se hace necesario desplazar las ondas de tensión y corriente modificando la

referencia de tierra a un valor que se corresponde al valor pico a pico (la amplitud de la onda) entre dos. Gráficamente lo que se pretende es lo siguiente:

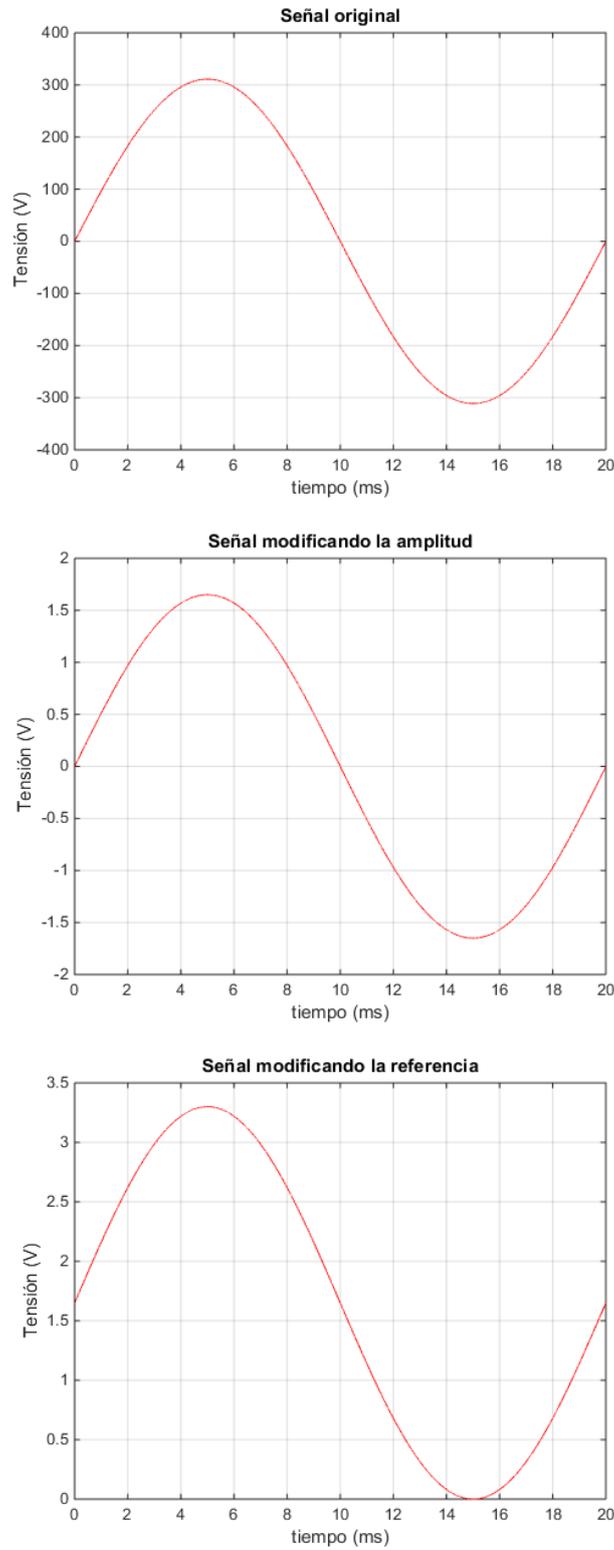


Figura 3. Ajuste de la señal de entrada a la instrumentación

Donde en un primer paso se adaptan las señales de tensión y corriente al rango de la instrumentación, y ya que se tiene la señal escalada al rango de la instrumentación se modifica la referencia de forma que al principio la onda oscila entre  $+V_{\text{pico}}$  y  $-V_{\text{pico}}$  y después de modificar la referencia oscila entre 0 y  $+2V_{\text{pico}}$ .

Este proceso de adecuación de las señales a la instrumentación se describe en 3 *Análisis de alternativas y descripción de la solución adoptada*, en lo que respecta a este capítulo, se describirá el funcionamiento del amplificador operacional, pues es una de las alternativas descritas para fijar un valor de tensión, así como los tipos de convertidores y los diferentes protocolos de transmisión de información entre el ADC y el microcontrolador.

## 2.1. Amplificador operacional

Como se ha descrito anteriormente, el amplificador operacional (AO) es una de las alternativas utilizadas para fijar la tensión de referencia, de forma que gráficamente, el esquema del sensor queda:

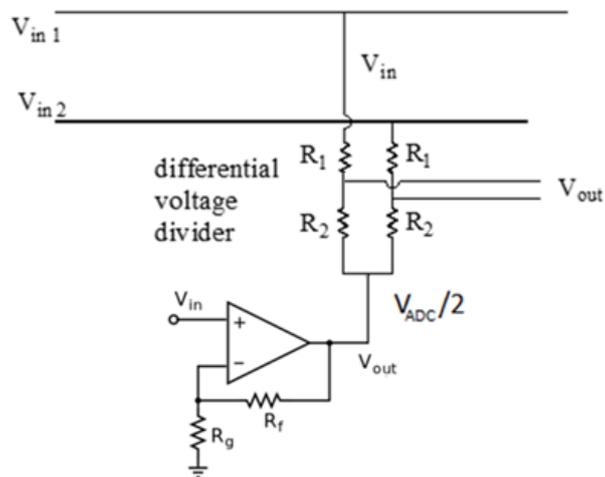


Figura 4. Sensor de tensión con una tensión de referencia distinto de tierra

En el presente capítulo se describirá en qué consiste y el modo de funcionamiento del amplificador operacional para ser utilizado en esta aplicación, mientras que en capítulos posteriores se discutirá el uso de ésta solución frente a otras.

De forma general, un amplificador operacional consta de dos entradas, una entrada no inversora ( $V_+$ ) y otra entrada inversora ( $V_-$ ). De forma ideal, el dispositivo amplifica la diferencia de tensión entre ambas entradas, esta tensión se denomina *tensión de entrada diferencial* ( $V_{\text{in}} = V_+ - V_-$ ), y la tensión de salida viene dada por la siguiente ecuación:

$$V_{\text{out}} = A_{Ao} \cdot (V_+ - V_-) \quad (\text{I.})$$

Donde  $A_{Ao}$  representa la ganancia del amplificador (cuando no hay realimentación, condición conocida como lazo abierto).

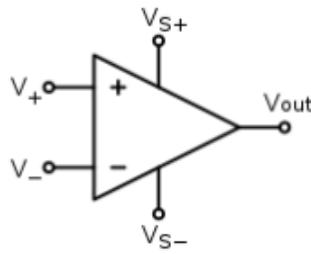


Figura 5. AO lazo abierto

Además de esta configuración, también existe el funcionamiento en lazo cerrado. Ambos modos de funcionamiento serán descritos en los siguientes apartados.

### 2.1.1. Lazo abierto

Un amplificador operacional ideal se caracteriza por tener una impedancia de entrada infinita, una impedancia de salida igual a cero y una ganancia  $A_{Ao}$  infinita. El hecho de que la impedancia de entrada sea infinita, implica que la corriente de entrada sea cero (en la práctica la impedancia de entrada es muy grande, lo que implica corrientes muy pequeñas).

El esquema de este tipo de funcionamiento se corresponde con el de la imagen anterior, y de la misma forma que antes, la tensión  $V_{out}$  se relaciona con la tensión de entrada mediante la siguiente expresión:

$$V_{out} = A_{Ao} \cdot (V_+ - V_-) \quad (II.)$$

Puesto que la ganancia es infinita, idealmente, si  $V_+$  es mayor que  $V_-$ , de la ecuación anterior se deduce que la tensión de salida tiende a ser infinita, sin embargo, puesto que se alimenta a una tensión  $V_s$ , la tensión de salida se saturará y será igual a  $+V_s$ , de forma análoga, si  $V_+$  es menor que  $V_-$ , la tensión de salida tenderá a ser menos infinito, pero por el mismo razonamiento, se saturará y será igual a  $-V_s$ . Gráficamente, si consideramos la entrada no inversora ( $V_+$ ) como una función senoidal con una amplitud de 3,3 V, y la entrada inversora ( $V_-$ ) como un valor constante e igual a cero y si se considera además que el amplificador se alimenta a una tensión de 3,3/2 V, la tensión de salida tendrá una forma de onda cuadrada cómo se representa a continuación:

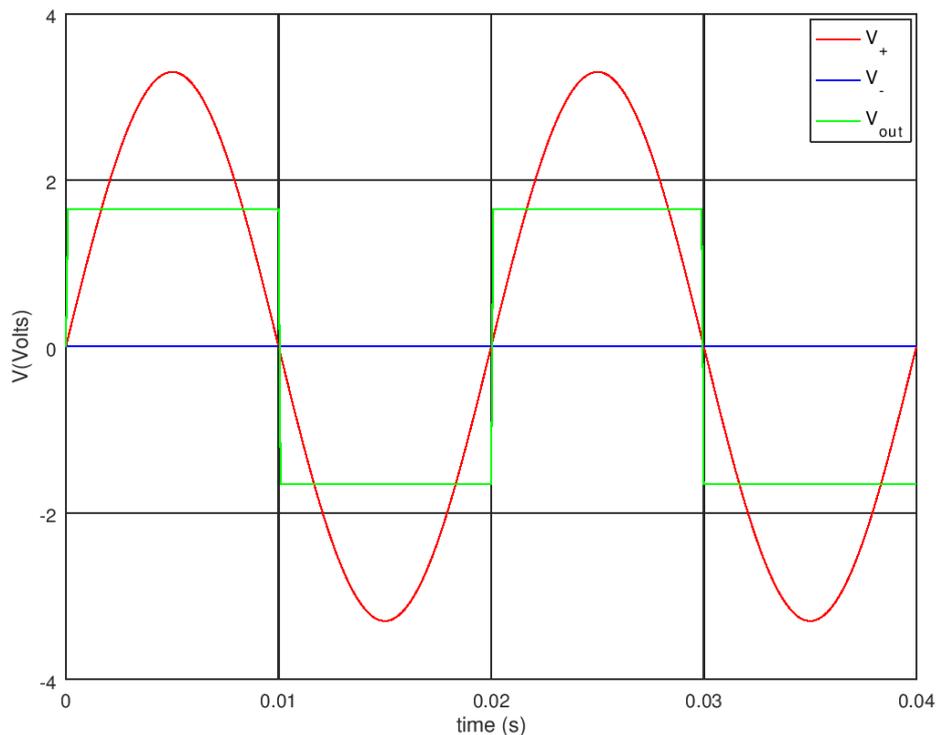


Figura 6. AO lazo cerrado señal de salida con una señal senoidal en entrada no inversora

Como se puede ver gráficamente, cuando la diferencia entre  $V_+$  y  $V_-$  toma valores positivos, la tensión de salida es  $+V_s$ , mientras que cuando la diferencia es negativa, la tensión de salida  $V_{out}$  es  $-V_s$ .

Notar que la aplicación de este tipo de configuración es muy limitada. Principalmente se usa como comparador. En lo que concierne a este proyecto, el fin que se busca con un amplificador operacional es fijar un cierto valor de tensión constante. Para ello, se tendrá una señal de entrada constante (la entrada no inversora tendrá un valor de tensión constante y distinta de cero, y la entrada inversora estará referida a masa), siendo la ganancia la constante de proporcionalidad que relaciona la tensión de entrada con la de salida. Para esto, se hace necesario tener una configuración en lazo cerrado, cuya configuración se describe a continuación.

### 2.1.2. Lazo cerrado

Como se ha visto en el apartado anterior, la configuración en lazo abierto implica que la tensión de salida será igual a la tensión de alimentación, y el signo de la tensión de salida dependerá del signo de la diferencia entre la tensión de entrada inversora y no inversora. Sin embargo, el objetivo que se persigue es tener una tensión a la salida proporcional a la entrada, y además, para simplificar el montaje, se pretende que la tensión de alimentación sea igual que la tensión de entrada. Gráficamente, una configuración en lazo cerrado es el siguiente:

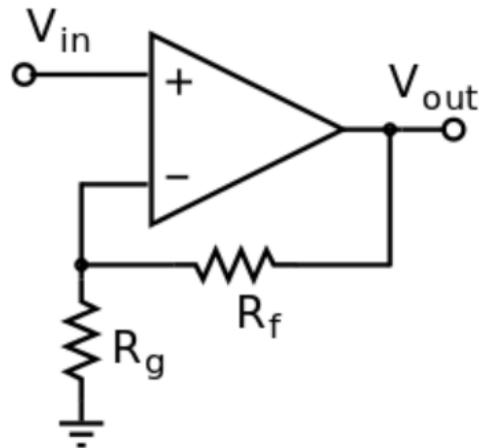


Figura 7. AO lazo cerrado

La tensión de salida viene dada por la siguiente expresión:

$$v_s = A_v \cdot v_e \quad (\text{III.})$$

Donde la tensión de entrada es la diferencia entre la tensión de entrada no inversora e inversora. De forma que se puede reescribir como:

$$v_s = A_{Ao} \cdot (V_+ - V_-) \quad (\text{IV.})$$

Notar que la tensión  $V_s$  es directamente la tensión de salida  $V_{out}$  (ambas referidas a masa)

$$v_s = V_{out} \quad (\text{V.})$$

Teniendo en cuenta que la corriente que circula por la resistencia  $R_f$ , sólo se puede cerrar por la resistencia  $R_g$  hacia tierra (puesto que la impedancia de entrada del amplificador operacional es muy grande, es decir, la corriente tanto de la entrada inversora como de la no inversora es cero), se puede calcular esta corriente como:

$$i = \frac{V_{out}}{R_f + R_g} \quad (\text{VI.})$$

Por lo que la tensión de entrada inversora será igual a (por la Ley de Ohm):

$$V_- = \frac{V_{out}}{R_f + R_g} \cdot R_g \quad (\text{VII.})$$

Si sustituimos esta expresión en(IV) tenemos:

$$V_{out} = A_{Ao} \cdot \left( V_+ - \frac{R_g}{R_f + R_g} \cdot V_{out} \right) \quad (\text{VIII.})$$

Despejando la tensión de salida tenemos:

$$V_{out} = \frac{V_+}{\frac{1}{A_{Ao}} + \frac{R_g}{R_f + R_g}} \quad (\text{IX.})$$

Teniendo en cuenta que la ganancia tiene a infinito, la tensión de salida será

$$V_{out} = \lim_{A_{Ao} \rightarrow \infty} \frac{V_+}{\frac{1}{A_{Ao}} + \frac{R_g}{R_f + R_g}} = \frac{V_+}{\frac{R_g}{R_f + R_g}} = V_+ \cdot \left( \frac{R_f + R_g}{R_g} \right) \quad (\text{X.})$$

Por lo que la relación entre la tensión de entrada y la de salida es:

$$V_{out} = V_{in} \cdot \left( \frac{R_f + R_g}{R_g} \right) \quad (\text{XI.})$$

Donde  $\frac{R_f + R_g}{R_g}$  será siempre mayor que uno, por lo que amplifica la señal de salida.

Notar que en este tipo de montajes la tensión de alimentación debe ser mayor que la tensión de entrada, en caso contrario el amplificador operacional siempre estará en modo saturación.

## 2.2. Tipos de convertidores ADC.

Para esta parte del presente capítulo se seguirá la misma estructura que la planteada en el libro de Instrumentación Electrónica de Miguel Ángel Pérez García<sup>(4)</sup>, puesto que presenta un desarrollo que resulta fácil de seguir incluso para aquellos lectores que tengan conocimientos básicos sobre electrónica e instrumentación. La información así como las imágenes incluidas en esta parte se ha extraído del libro anteriormente mencionado, del libro Exploring Arduino de Jeremy Blum<sup>(5)</sup>, así como de diversas fuentes de internet<sup>(6)(7)(8)</sup>.

Los convertidores ADC se pueden clasificar desde varios puntos de vista: según sea el tipo de entrada, de salida, la forma de hacer la conversión, la forma de codificación de la variable digital de salida, etc. En las siguientes secciones, se describirá los diferentes tipos de convertidores en función de las principales clasificaciones que se pueden distinguir<sup>(6)</sup>.

### 2.2.1. Entrada diferencial y entrada referida a masa

Desde el punto de vista del tipo de entrada, se puede afirmar que la mayoría de los convertidores ADC trabajan con señales de tensión, sin embargo, existen dos opciones básicas para recibir la señal de entrada de tensión.

Una de las opciones consiste en disponer de un terminal referido a masa y el otro terminal con la señal que se desea medir. Este tipo de entrada también se conoce como single ended. Gráficamente consiste en lo siguiente:

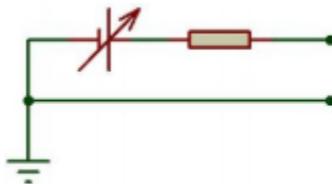


Figura 8. Entrada single ended<sup>(6)</sup>

En este caso, al estar referido a masa, la señal de entrada será igual a la tensión del terminal

Otra de las opciones es el caso de medir en la entrada una señal diferencial.

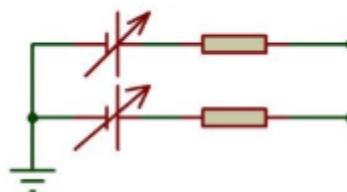


Figura 9. Entrada diferencial<sup>(6)</sup>

En este caso, la señal de entrada será la diferencia entre el valor de tensión de ambos terminales. Notar que en este caso, la señal de tensión en cada terminal debe ser menor que la tensión de saturación de la instrumentación, pues aunque la señal de entrada es la diferencia entre ambos terminales, para poder medir la tensión de cada terminal, ésta debe estar dentro del rango de la instrumentación para cada terminal

Existe una combinación entre los dos tipos de entrada anteriores, que es la pseudo-diferencial, que consiste en que la señal de referencia, en lugar de estar referida a masa, está referida a una tensión distinta, pero constante.

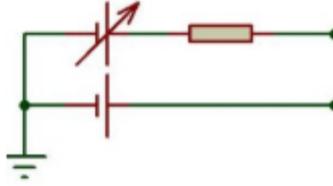


Figura 10. Entrada pseudo-diferencial<sup>(6)</sup>

Así pues, estas alternativas se muestran gráficamente a continuación:

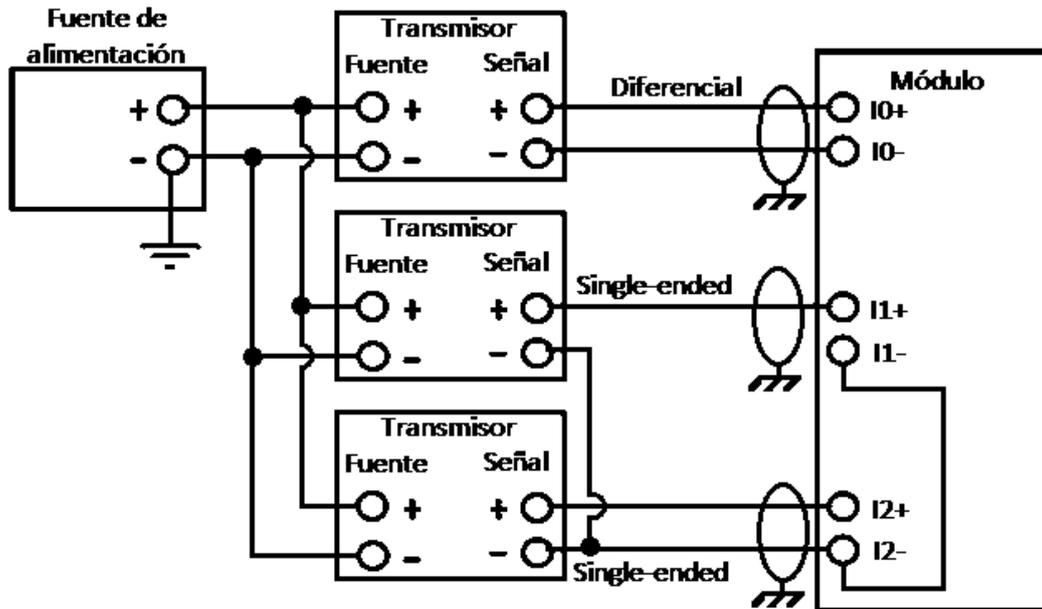


Figura 11. Tipos de entradas<sup>(9)</sup>

Notar que, en una entrada del tipo single-ended, no se lleva el terminal referido a masa porque internamente el módulo lleva este terminal a masa.

Además, se puede clasificar también según el tipo de conexión a masa:

- Grounded: la masa de la señal es igual a la del sistema.
- Flotante: la masa de la señal no coincide con la del sistema.

### 2.2.2. Salidas en serie y en paralelo

Desde el punto de vista del tipo de salida, se puede agrupar en dos grandes categorías: las salidas en paralelo o en serie.

En el caso de la salida en paralelo, las señales que se corresponden a la variable digital de salida del convertidor se transmiten de forma simultánea, mientras que en el caso de las salidas en serie, los valores de cada señal se suministran de forma secuencial mediante una única salida.

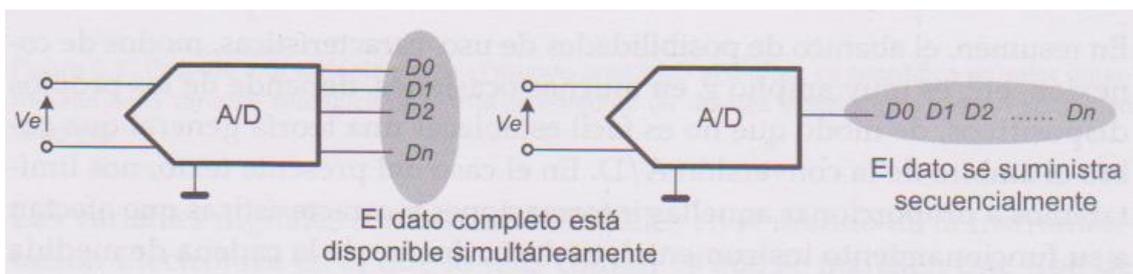


Figura 12. Salida en serie y paralelo<sup>(6)</sup>

La principal ventaja de la salida en paralelo es su sencillez ya que todos los bits (datos) se transmiten de forma simultánea sobre varias líneas, estando de este modo disponible directamente la información sobre las líneas digitales, por esto motivo, no es necesario realizar ninguna operación para leerlo; el principal inconveniente es que precisa de tantas líneas como señales digitales de salida se tengan lo que aumenta el costo y la complejidad del sistema digital. En este tipo de salidas digitales será necesario disponer por lo tanto de tantos pines en el convertidor y en el microcontrolador como número de bits a transmitir (líneas).

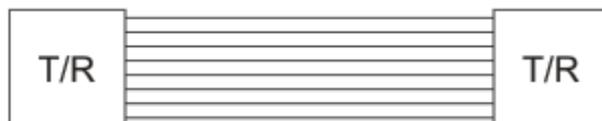


Figura 13. Salida paralelo

Por otro lado, los convertidores con salida serie emplean muy pocas líneas de interconexión (una para los datos más las necesarias para el control); sin embargo, son intrínsecamente más lentos, puesto en los de salida en paralelo, únicamente se tiene un tiempo de transmisión entre cada bloque de datos, mientras que en el caso de salida en serie, sobre la misma línea se envían todos los bits de información, lo que ya de por sí retrasa la transmisión de información, pero además, se precisan de algún tipo de operación para obtener el dato final. Debido a las elevadas velocidades de transmisión de hoy en día este factor no resulta crítico para la mayoría de las aplicaciones, y debido a la existencia de varios protocolos normalizados se hace sencillo manejar la información del convertidor.

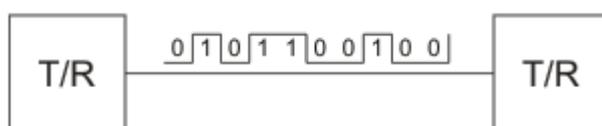


Figura 14. Salida serie

Los protocolos de comunicación serie más comunes son el SPI (Serial Peripheral Interface), también es conocido como SSI (synchronous Serial Interface) debido a la característica síncrona de su comunicación. Otro de los protocolos más habituales es la comunicación I<sup>2</sup>C (Inter-Integrated Circuit), un formato desarrollado por Philips que es de uso libre desde 2006. Una particularización de este protocolo es el I<sup>2</sup>S (Inter-IC Sound) que se emplea a menudo en la comunicación de convertidores ADC en aplicaciones de audio.

### 2.2.3. Codificación del dato de salida

La codificación es la representación numérica de los valores al pasarlos de continuos (señal analógica) a discretos (señal digital).

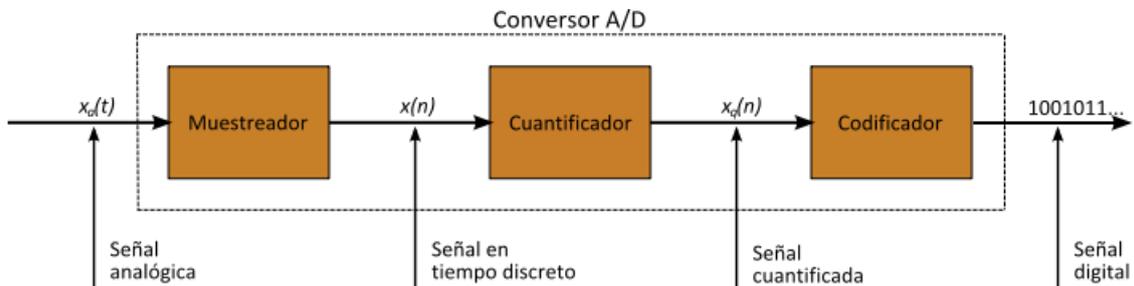


Figura 15. Esquema convertidor ADC

El código más utilizado es el código binario puesto que permite una expresión sencilla de la cuantificación del valor de la variable y en este formato es como se suele trabajar con los datos en los sistemas microprocesadores y microcontroladores; sin embargo existen otros como el código Gray o los formatos decimales codificados en binario (BCD).

En la actualidad, puesto que la capacidad de gestión de los microprocesadores es inmensa, se suele usar el binario.

De esta forma, con la codificación se pretende asignar un nivel de cuantificación a la señal analógica. Para el caso de convertidores, cuanto mayor sea la resolución, mayor será el número de bits, por lo que se tendrá un mayor número de niveles de codificación y el error será menor, puesto que la señal digital se aproxima más al valor analógico.

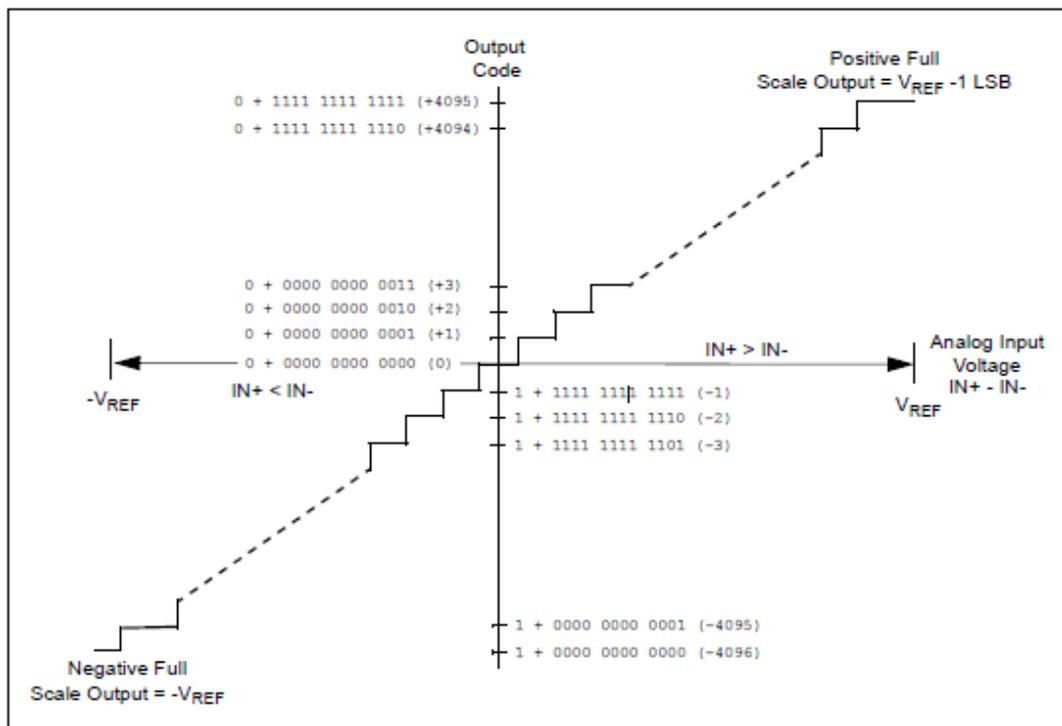


Figura 16. AO Codificación ADC: asignación de nivel de cuantificación (salida digital) en función de la señal analógica (entrada analógica)

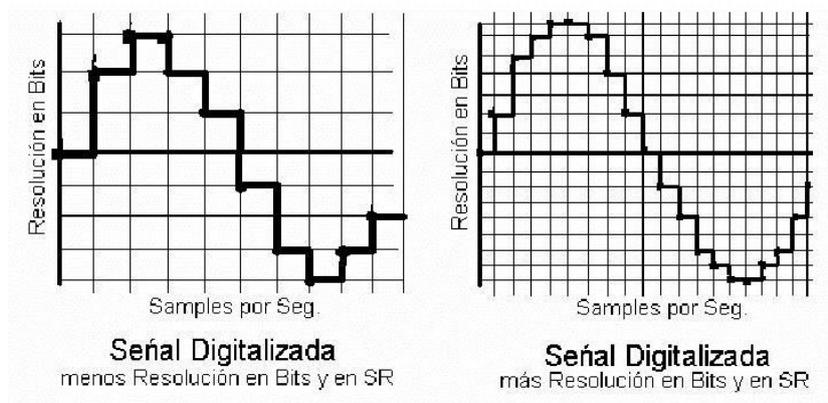
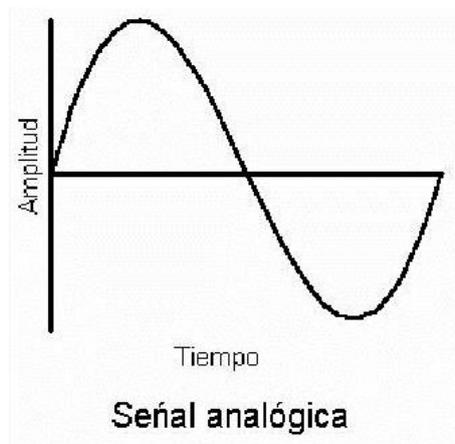


Figura 17. Salida analógica con diferente resolución (8 y 16 bits)

### 2.2.4. Topología de conversión

Como ya se ha ido describiendo en las secciones anteriores, las estructuras de los convertidores proporcionan un conjunto de señales digitales (bits) cuyo código represente el valor de la variable analógica de entrada.

La forma más evidente de realizar una conversión de analógico a digital es mediante el uso de comparadores y un codificadorlo que constituye un convertidor ADC directo (flash converter). Esta estructura se muestra en la siguiente imagen:

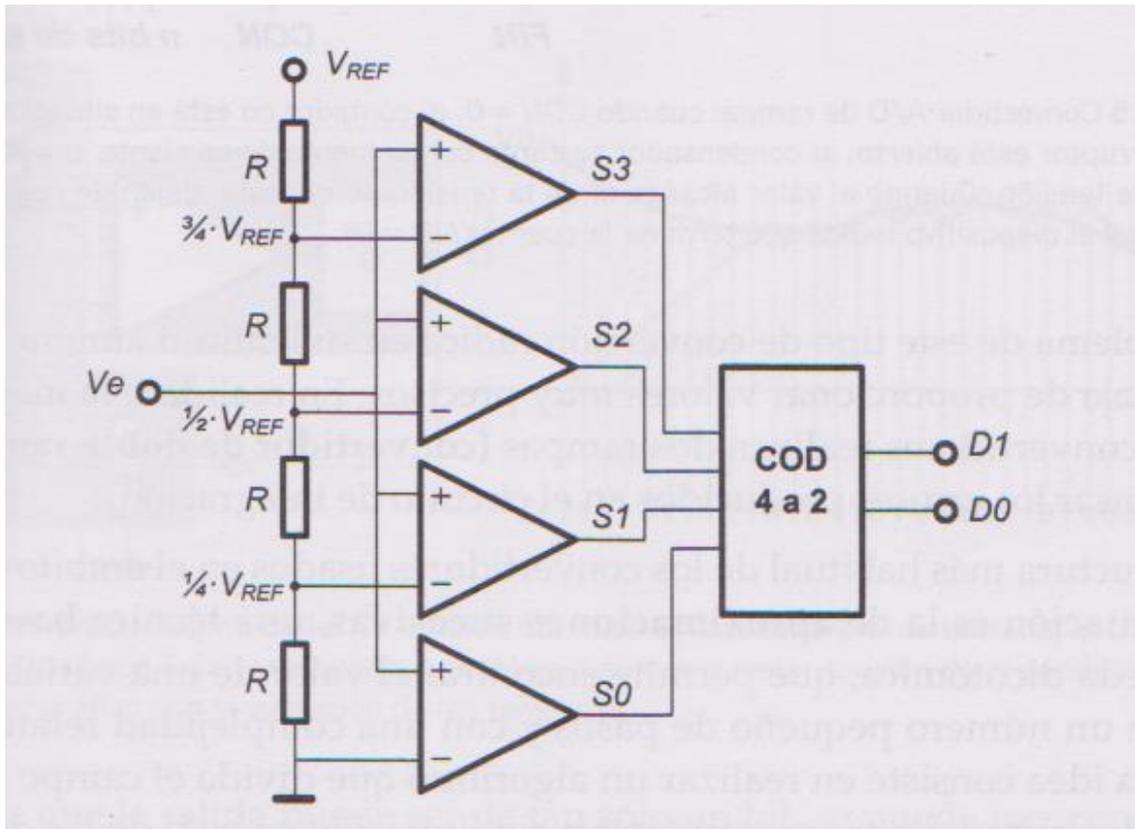


Figura 18. Conversión Flash converter<sup>(6)</sup>

La forma de funcionamiento es la siguiente:

- De cada comparador se tiene una señal digital que indica si la señal analógica está situada por encima o por debajo de su nivel de comparación.
- El conjunto de todas estas señales se codifica con un circuito digital combinacional para dar el código correspondiente.

Este tipo de estructura es poco eficiente puesto que en aplicaciones donde se desee una mayor resolución y se incremente el número de bits, provoca un aumento en el número de comparadores, y por tanto, en coste. Por ejemplo, para conseguir 8 bits, se precisaría contar con  $2^8 - 1 = 255$  niveles, y para comparar el valor analógico con el nivel de cuantificación correspondiente, será necesario este mismo número de comparadores. La principal ventaja de este convertidor es que puede ser muy rápido, tanto como lo sean los comparadores.

Otra forma de realizar la conversión de analógico a digital es mediante convertidores de rampa. Esta solución se muestra en la siguiente imagen:

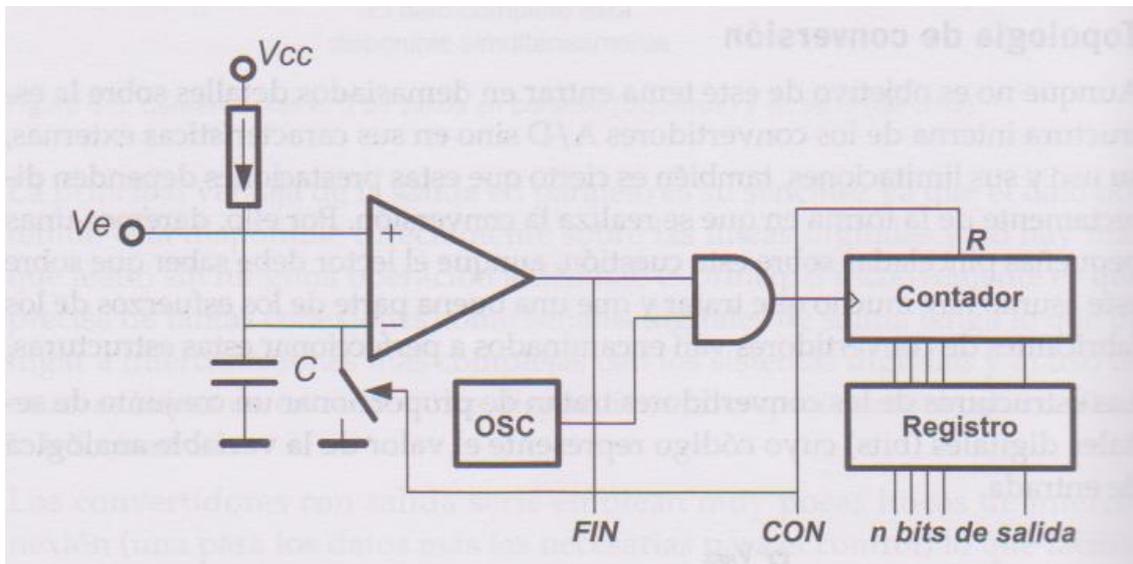


Figura 19. Conversión mediante rampas<sup>(6)</sup>

Estos dispositivos tienen un integrador que aumenta la tensión hasta alcanzar el valor de la variable analógica de la entrada; mientras tanto, un contador cuenta pulsos hasta el momento en que la rampa y la señal de entrada tengan el mismo valor, de forma que el valor numérico del nivel de cuantificación es proporcional al número de pulsos contados, y el número de pulsos contado dependerá del tiempo que tarde la rampa en igualar la señal de entrada, así pues, un mayor tiempo para igualar la señal de entrada, implicará un mayor número de pulsos contados, y por lo tanto, un mayor nivel de cuantificación.

El principal problema de este tipo de conversión es su lentitud, aunque proporciona valores muy precisos. En la práctica, la mayor parte de los convertidores realizan dos rampas (convertidor doble rampa) para compensar los errores producidos en el circuito de integración.

Las estructuras más habituales son: aproximaciones sucesivas y sigma-delta

#### 2.2.4.1. Aproximaciones sucesivas

A groso modo, esta técnica consiste en realizar comparaciones sucesivas de manera descendente o ascendente para rechazar rangos de voltajes, hasta que se encuentra la combinación que iguala la tensión entregada por el convertidor y la entrada.

En concreto, consiste en realizar un algoritmo que divide el campo de medida en dos mitades, para a continuación descartar el campo donde no está el valor de la variable. En el siguiente paso repite el mismo proceso, por lo que la zona se reduce por cuatro con respecto al original. Los pasos sucesivos reducen la zona una octava, dieciseisava parte, etc., hasta llegar a la precisión que se desee. El número de pasos es igual al número de bits de cuantificación con lo que este tipo de sistema supone una solución de compromiso entre la rapidez de los convertidores directos y la precisión de los convertidores de rampa.

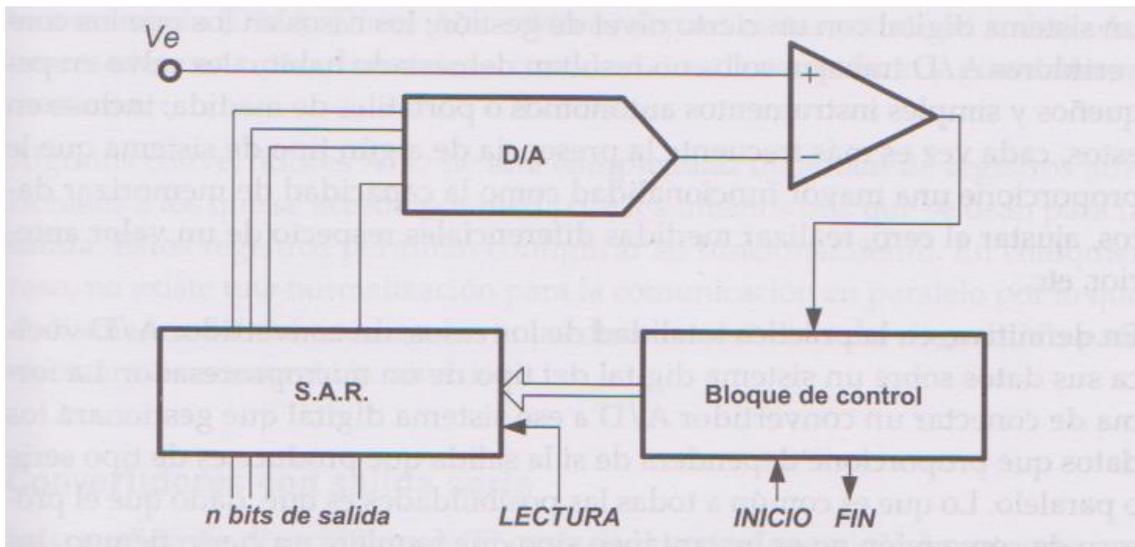


Figura 20. Conversión mediante aproximaciones sucesivas<sup>(6)</sup>

#### 2.2.4.2. Sigma-delta

Este tipo de convertidor, se basa en comparar el valor de la variable de entrada con un valor almacenado que aumenta o disminuye en función de la comparación. El resultado de la comparación determina si en el siguiente paso se incrementa o se disminuye el valor almacenado. También se suele denominar convertidor de 1 bit ya que el resultado solo indica si el valor sube o baja respecto del anterior.

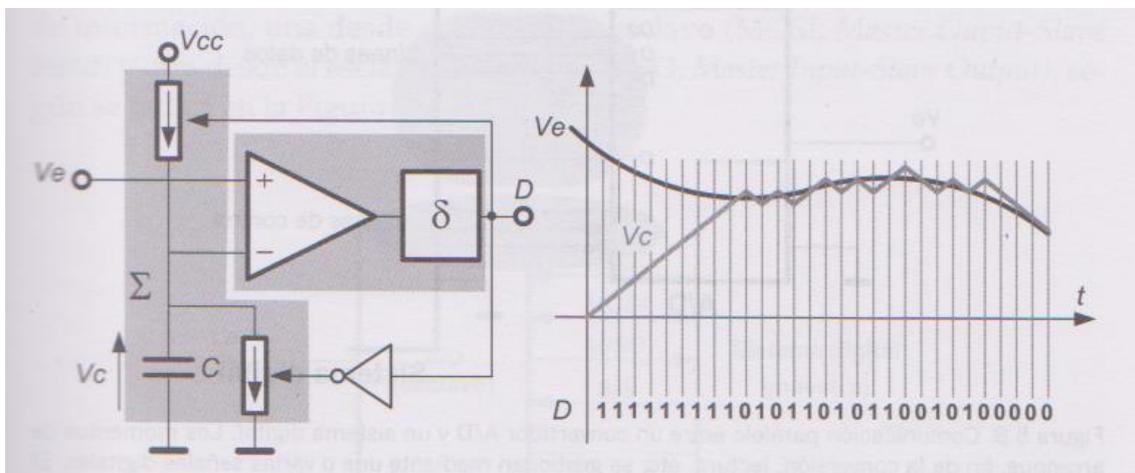


Figura 21. Conversión sigma-delta<sup>(6)</sup>

### 2.3. Conexión de convertidores ADC a sistemas digitales

En escasas aplicaciones los convertidores ADC funcionan de forma aislada salvo en pequeños y simples instrumentos autónomos o portátiles a medida (incluso en estos casos, cada vez es más frecuente la presencia de algún tipo de sistema que le proporcione una mayor funcionalidad como la capacidad de memorizar datos, ajustar el cero, realizar medidas diferenciales respecto de un valor anterior, etc.); de forma que en la mayoría de los casos, los convertidores forman parte de una cadena de control o de monitorización, donde los datos que obtienen se transmiten a un sistema digital que tendrá un cierto nivel de gestión gracias a los datos recibidos.

La forma de conexión de un convertidor ADC a ese sistema digital dependerá exclusivamente de si la salida que produce es de tipo serie o paralelo.

#### 2.3.1. Convertidores con salida en paralelo

El principio de funcionamiento se vio en 2.2.2 *Salidas en serie y en paralelo*. Esta forma de conexión es cada vez menos habitual, debido a que requiere de muchas líneas para transmitir los datos, tantas como bits se empleen para la conversión más aquellas dedicadas a la gestión del diálogo.

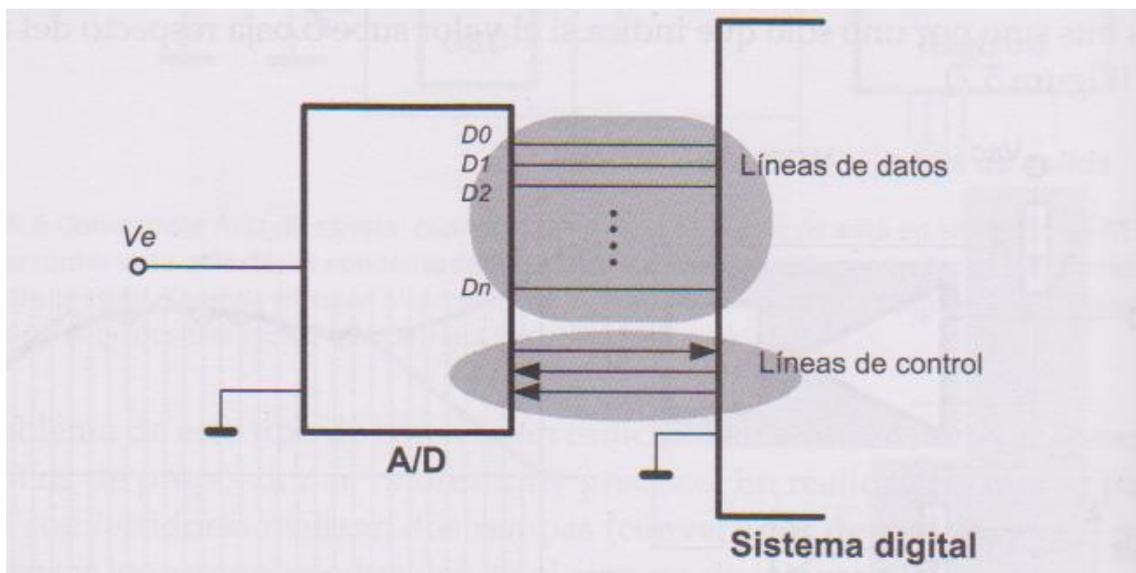


Figura 22. Salida paralelo<sup>(6)</sup>

En general, la comunicación se gestiona mediante algún tipo de señal de control que indica los momentos de inicio y fin de conversión.

Algunos convertidores ADC de alta complejidad disponen de registros adicionales a los que se accede por medio de los mismos bits que se usan para la salida. Estos registros permiten configurar su funcionamiento. En cualquier caso, no existe una normalización para la comunicación en paralelo por lo que el diseñador deberá acudir a las hojas de características del dispositivo para efectuar correctamente la conexión.

### 2.3.2. Convertidores con salida serie

La conexión de dispositivos ADC con salida serie depende del protocolo implementado en el convertidor que suele ser de tipo SPI o I<sup>2</sup>C, pero está estandarizada en cualquiera de los casos.

La conexión SPI no es un estándar propiamente dicho, pero está tan extendido su uso que se puede considerar como un estándar de hecho. Este tipo de comunicación utiliza líneas separadas para enviar y recibir información, lo que lleva a que requiere de líneas adicionales con respecto al I<sup>2</sup>C. Se trata de un protocolo de comunicación síncrono (un reloj controla la transferencia de datos) de tipo full-duplex (se permite la comunicación bi-direccional de forma simultánea) entre un sistema que actúa como maestro (master) y uno o varios que funcionan como esclavos (slaves). En el caso de conexión de uno o varios convertidores ADC a un sistema digital, este será el maestro mientras que los esclavos serán los convertidores ADC. La conexión física se realiza mediante cuatro señales referidas a masa, de las cuales una es el reloj de sincronización SCLK, que será proporcionado por el equipo digital que actúa como maestro, una línea de selección de esclavo CS ó SS y dos líneas para la transmisión de información, una desde el maestro al esclavo (MOSI, Master Output-Slave Input) y otra desde el esclavo al maestro (MISO, Master Input-Slave Output). De forma que para la comunicación entre el maestro y el esclavo, se da mediante las líneas de selección de esclavo (CS), y las líneas Master Out Slave In y Master In Slave Out, siendo la línea de sincronización la que marca el instante de tiempo donde se da el trasiego de información.

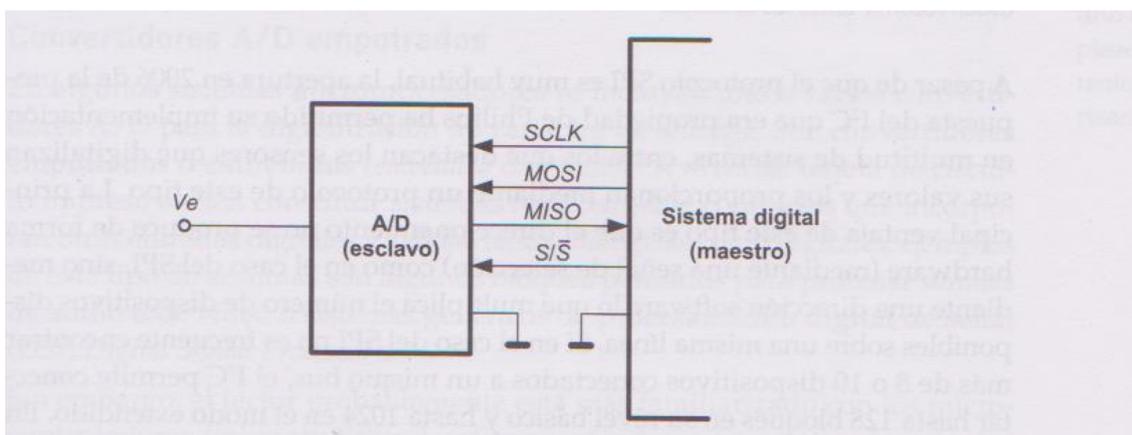


Figura 23. Comunicación SPI con un esclavo<sup>(6)</sup>

Cada esclavo requiere además de una línea adicional, que es el Slave Select (CS). Por lo tanto, el número total de pines de I/O requeridos será siempre 3+n, siendo n el número de esclavos. La siguiente figura muestra la configuración de este protocolo:

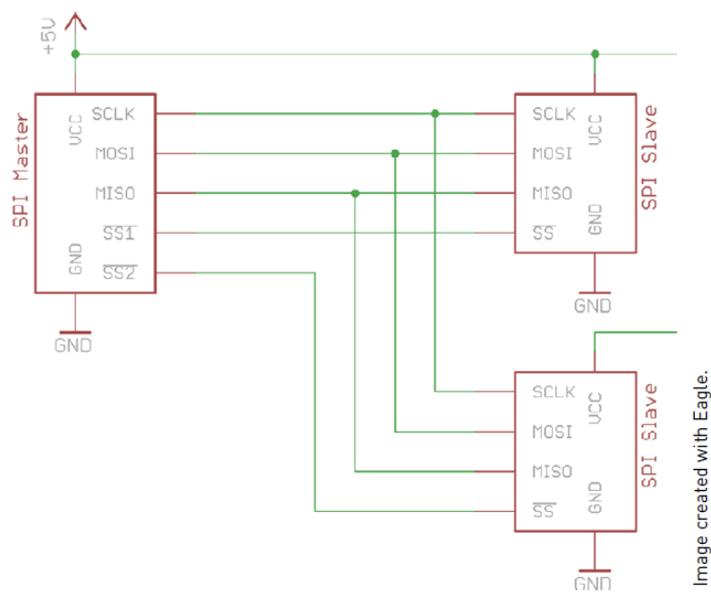


Figura 24. Comunicación SPI con más de un esclavo

En el caso de que se deseen conectar varios convertidores ADC como, por ejemplo, cuando se quiere conectar un determinado número de sensores independientes cuya salida es del tipo SPI, existe otra configuración para ahorrar pines. La que se ha mostrado anteriormente, consiste en una estructura de esclavos independientes en la que el sistema digital selecciona en cada momento el convertidor con el que va a entablar el diálogo mediante una línea específica. La segunda forma de hacerlo es un tipo de priorización por posición conocida como daisychain en la que sólo cuando el más cercano de los convertidores ADC está libre, pueden hablar los demás, una solución de gestión más compleja pero que reduce a sólo una línea de selección las necesidades del sistema digital que actúa como maestro.

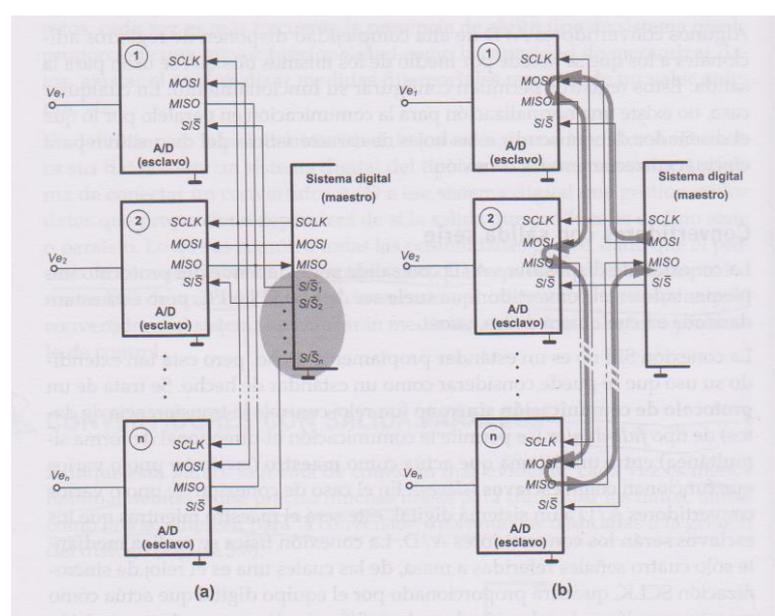


Figura 25. Comunicación SPI con más de un esclavo - daisychain<sup>(6)</sup>

Así pues, cuatro líneas, al menos, están presentes en una configuración SPI, cuya funcionalidad es, para cada uno:

- MOSI: se usa para enviar la información desde el dispositivo maestro al dispositivo que funciona como esclavo.
- MISO: se usa para enviar la información desde el dispositivo esclavo al dispositivo que actúa como maestro.
- SCLK: consiste en la señal, mediante la cual se sincroniza la información que se envía al dispositivo receptor desde el emisor, y del mismo modo, el receptor lee la información referida a éste.
- SS: línea dedicada a la selección del dispositivo esclavo. Por defecto tiene un valor alto, por lo que un valor bajo indica que la comunicación se va a establecer con el esclavo conectado a esa línea.

A pesar de que el protocolo SPI es muy habitual, la apertura en 2006 de la propuesta del I<sup>2</sup>C que era propiedad de Philips ha permitido su implementación en multitud de sistemas, entre los que destacan los sensores que digitalizan sus valores y los proporcionan mediante un protocolo de este tipo. La principal ventaja de este tipo es que el direccionamiento no se produce de forma hardware (mediante una señal de selección) como en el caso del SPI, sino mediante una dirección software lo que multiplica el número de dispositivos disponibles sobre una misma línea. Si en el caso del SPI no es frecuente encontrar más de 8 o 10 dispositivos conectados a un mismo bus, el I<sup>2</sup>C permite conectar hasta 128 bloques en su nivel básico y hasta 1024 en el modo extendido. En realidad, dado que algunas de las direcciones están reservadas para determinadas funciones no llegan a ser tantas, pero un sistema instrumental que maneje ese número de variables analógicas procedentes de otros tantos sensores es realmente un sistema complejo...

El protocolo I<sup>2</sup>C es muy sencillo y robusto desde el punto de vista de hardware ya que se compone únicamente de dos líneas de colector abierto (open-collector) o drenaje abierto (open-drain), una para la comunicación (SDA) y otra para actuar como reloj de sincronismo (SCL). Cuando el bus no está siendo usado las dos líneas están a nivel alto y cualquier maestro puede acceder al bus poniendo a nivel bajo la línea SDA, luego pondrá la dirección del receptor y, finalmente, se establecerá el diálogo que terminará con la vuelta de la línea SDA a nivel alto. Por esto, una de las ventajas de este protocolo es que utiliza un mínimo número de pines I/O para facilitar la comunicación.

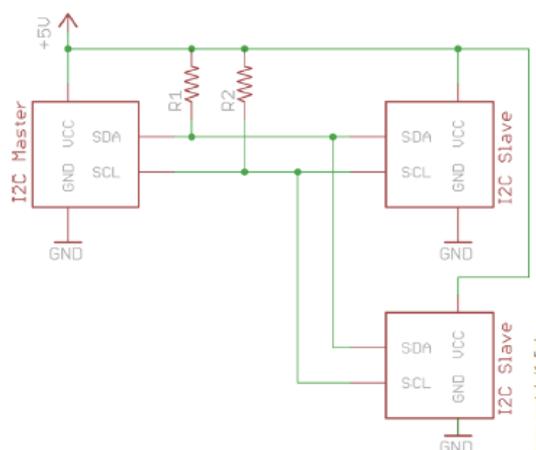


Figura 26. Comunicación I<sup>2</sup>C

El protocolo I<sup>2</sup>C permite que varios esclavos compartan las líneas de comunicación con un único dispositivo maestro. Los esclavos no pueden iniciar la comunicación, únicamente pueden responder a la señal recibida por el maestro. Debido a que los esclavos comparten la misma línea de comunicación, es muy importante que sólo el maestro pueda iniciar la comunicación, en caso contrario, diversos esclavos podrían intentar enviar señales al mismo tiempo resultando en una información ilegible.

Todos los comandos y peticiones enviados por el maestro, son recibidos por todos los esclavos. Cada esclavo I<sup>2</sup>C tiene una única dirección de 7-bit, o número ID. Este dirección ID propia para cada esclavo, no era necesario en el protocolo SPI, puesto que tiene una línea adicional para seleccionar el esclavo que debe responder a la información que lleva del maestro. Cuando se inicia la comunicación por el maestro, se transmite del mismo modo la ID del dispositivo. El esclavo únicamente responde a la información recibida, si la dirección ID asociada se corresponde al mismo. Puesto que todos los dispositivos reciben todos los mensajes, al compartir el mismo bus, cada dispositivo debe tener una dirección ID única.

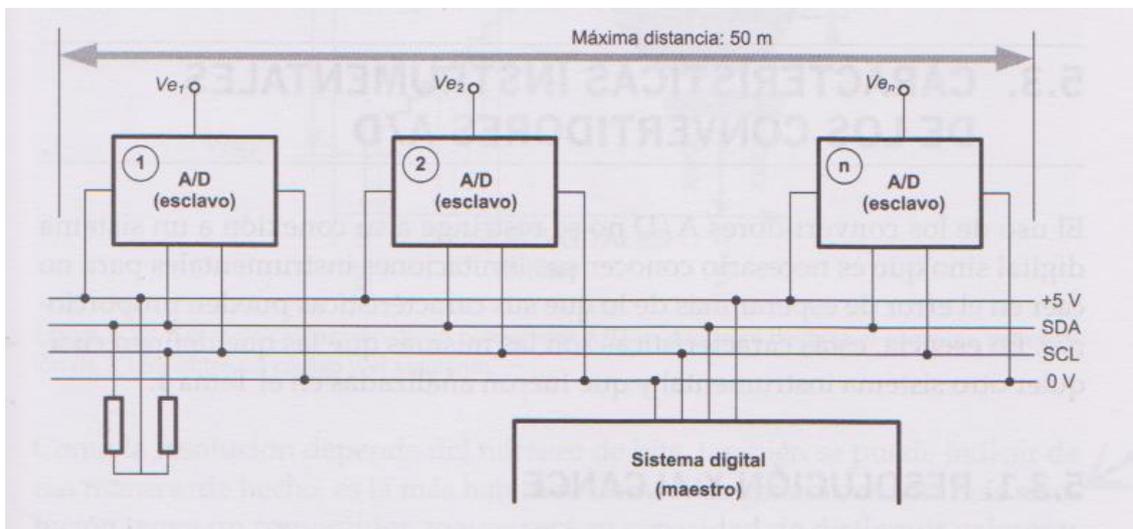


Figura 27. Comunicación I<sup>2</sup>C<sup>(6)</sup>

El protocolo I<sup>2</sup>C es muy flexible, pero presenta limitaciones en cuanto a su velocidad de transmisión de datos, que es reducida (en el mejor de los casos, 3,4 Mbit/s aunque el valor estándar es de 100 kbit/s), lo que no es un problema para la mayoría de los sistemas de instrumentación pero sí para otras aplicaciones dentro del mundo de la Informática. Sin embargo, la principal limitación para los sistemas instrumentales es su corto alcance –no más allá de 50 m o menos si el entorno es electromagnéticamente agresivo- lo que reduce su aplicación en el control de procesos y en otras aplicaciones industriales sólo a manejar dispositivos dentro de un mismo equipo o, incluso, dentro de una misma tarjeta de circuito impreso.

Así pues, una comparativa entre el SPI, y el I<sup>2</sup>C, permite concluir que:

- La configuración SPI permite operar con mayores velocidades, siendo además más fácil de trabajar con este tipo de conexiones. No requiere de resistencias y Arduino soporta a nivel de hardware este tipo de comunicaciones.
- Este tipo de comunicación requiere únicamente de dos líneas, lo que permite una mayor “economía de pines”. Del mismo modo que la configuración SPI, Arduino soporta este tipo de comunicaciones.

### 2.3.3. Convertidores A/D empotrados

En algunos sistemas microprocesadores se incluyen uno o varios convertidores A/D para la digitalización de variables de entrada. Son convertidores empotrados o embebidos (*embeddedconverters*). A nivel de tarjeta de circuito impreso suelen constituir sistemas relativamente complejos que incorporan otras entradas digitales y salidas tanto analógicas como digitales; ejemplos de este tipo de sistemas son algunos bloques pensados para procesar señales de audio o de vídeo o sistemas genéricos de procesamiento digital de señal (*DSP*, *Digital SignalProcessor*).

Sin embargo, el diseñador de instrumentación probablemente está más familiarizado con los microcontroladores (*microcontrollers*), microprocesadores especializados que incluyen en el mismo circuito integrado todos los dispositivos periféricos que podemos necesitar, tales como memoria, puertos de entrada/salida, comunicaciones serie y, por supuesto, convertidores A/D.

En estos casos, el conexionado ya está hecho en el propio chip y el usuario se limitará a programar adecuadamente el sistema para conseguir la funcionalidad buscada. Lógicamente, la forma de trabajar de estos sistemas dependerá del fabricante y a sus especificaciones habrá de dirigirse el diseñador cuando quieran emplearlos.

### 3. Análisis de alternativas y descripción de la solución adoptada

En el capítulo anterior se han descrito los aspectos teóricos que se desarrollarán a lo largo del documento. De esta forma, en este capítulo, se hará énfasis en las diferentes alternativas y se realizará la selección de la solución adoptada. El desarrollo de la solución final se describirá en los posteriores capítulos, siendo este capítulo un análisis descriptivo de las posibles soluciones, sin entrar en detalle de cómo se desarrollan todas las alternativas.

Así pues, es necesario desarrollar sensores de medida de tensión y corriente, mediante los cuales se determinarán el resto de magnitudes. Las diferentes alternativas se presentan a continuación.

#### 3.1. Selección de sensores

En cuanto a sensores de tensión, las posibles alternativas son<sup>(4)(10)</sup>:

- Divisor de tensión (diferencial): en cuanto a precio, esta alternativa es la más interesante, pues se compone únicamente de 4 resistencias. Asimismo, se obtiene una precisión buena con este tipo de sensor, además de ser simple.

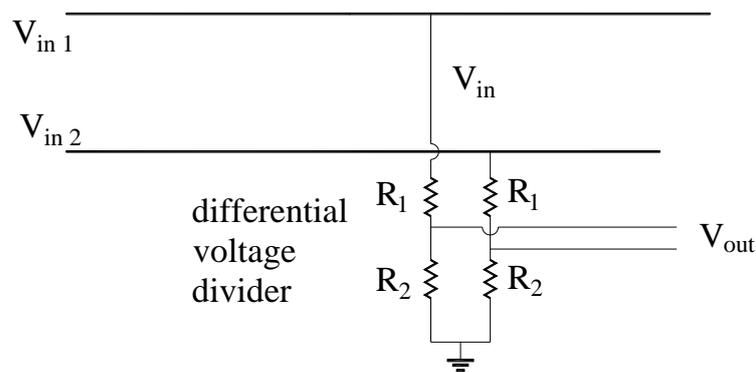


Figura 28. Divisor de tensión diferencial

Notar que debido a que el valor de las resistencias puede variar entre una y otra serie de fabricación dentro de los márgenes establecidos por la tolerancia especificada, para minimizar el error será necesario compensar estas desviaciones en el valor de las resistencias durante su montaje, seleccionando resistencias con valores cercanos a los especificados.

Una de las limitaciones de este tipo de sensores es que no presenta aislamiento, aunque para esta aplicación no será necesario pues se trabaja en baja tensión.

- Transformador de tensión: por precio, este es menos competitivo que el anterior. Consiste en un transformador de pequeña potencia que trabaja en vacío. Además del precio, en torno a 4€<sup>(11)</sup>, tiene el inconveniente de tener una peor respuesta para frecuencias diferentes a las que ha sido definido el transformador, esto es, que para frecuencias distintas a 50 Hz (en el caso de España) la impedancia variará, modificando de esta forma la caída de tensión. De esta forma, frecuencias mayores a la nominal implican una mayor impedancia y por lo tanto una mayor caída de tensión, y viceversa, una menor frecuencia implica una menor impedancia y por lo tanto una menor caída de tensión. Sin embargo, para esta aplicación, este no es un factor decisivo al medirse la tensión de red, donde la frecuencia es constante y la presencia de armónicos es despreciable. Otro inconveniente de este tipo de transformadores es que tienen una

corriente de vacío y una caída de tensión en el primario relativamente elevadas. Ambos efectos producen un error de fase significativo. Aunque la caída de tensión sería fácil de compensar para medidas exclusivamente de tensión, el error de fase hace que esta opción sea poco adecuada para medidas de tensión que posteriormente van a ser la base de cálculos de potencia eléctrica, lo que hace que sean muy poco adecuados para el objetivo de este proyecto.

En cuanto a sensores de corriente, tenemos:

- **Sensor Hall (lazo cerrado):** tiene varias ventajas como una gran precisión además de presentar una relación casi lineal con la corriente. Sin embargo, su precio está en torno a 10-30€<sup>(12)</sup>, y para conseguir la señal de salida requiere de alimentación externa, lo que lo hace menos robusto que la solución que se presenta a continuación.
- **Transformador de corriente:** esta solución es más económica, pues su precio está en torno a 2-3€<sup>(13)</sup> para el rango de potencia de nuestra aplicación. Asimismo, es un equipo más robusto que el anterior. Se basa en un transformador trabajando prácticamente en cortocircuito en cuyo devanado secundario tenemos instalada, por tanto, una resistencia muy pequeña, donde medimos una tensión que por la ley de Ohm será proporcional a la corriente que circula por la resistencia. Esta corriente, a su vez, al estar el transformador en cortocircuito, es proporcional a la corriente primaria.

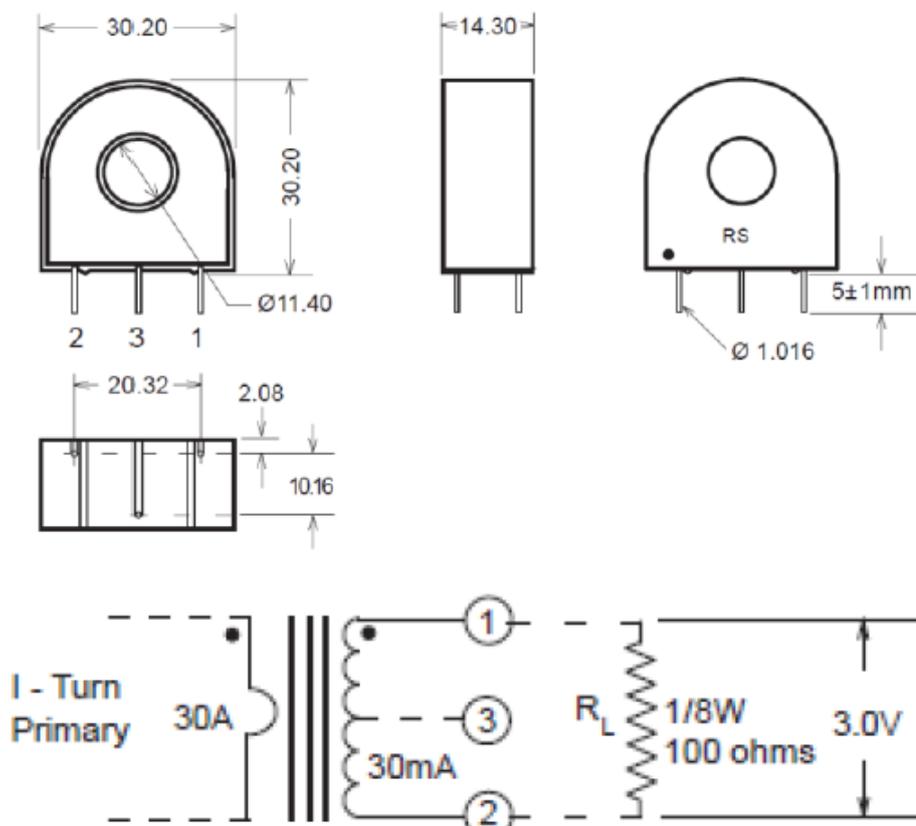


Figura 29. Transformador de corriente

Como el transformador de tensión, tiene el inconveniente de la peor respuesta en frecuencia, aunque, para esta aplicación, al utilizarse para medir corrientes a la frecuencia de red, este no es un factor decisivo.

Aunque la corriente consumida por muchos de los receptores domésticos presenta un contenido de armónicos elevado, el ancho de banda limitado del transformador de corriente tiene una incidencia marginal en la precisión de la medida de potencia. Esto se debe a que en la potencia interviene también la tensión de alimentación, con un contenido de armónicos muy inferior al de la corriente.

Los sensores utilizados para esta aplicación serán, un divisor de tensión diferencial, para adaptar la señal de tensión, y un transformador de corriente, para adaptarla señal de corriente.

### 3.2. Tensión de referencia

Por otro lado, notar que, el valor instantáneo de las magnitudes de tensión y corriente medidas entre los terminales que serían la entrada al ADC, tomarán valores negativos durante medio ciclo, esto es 0,01s (50 Hz), y positivos durante la otra mitad, es decir, los restantes 0,01s hasta completar un ciclo de onda (0,02s). Esta circunstancia no se puede dar, puesto que el convertidor se saturaría y no mostraría los valores reales. Es por lo tanto necesario cambiar la señal de referencia de ambas soluciones, de forma que la tensión en cada uno de los terminales oscile entre la tensión de saturación del convertidor y 0. La referencia se fijará por lo tanto a la mitad de la tensión a la cual el convertidor satura. Gráficamente, lo que se pretende conseguir es lo siguiente:

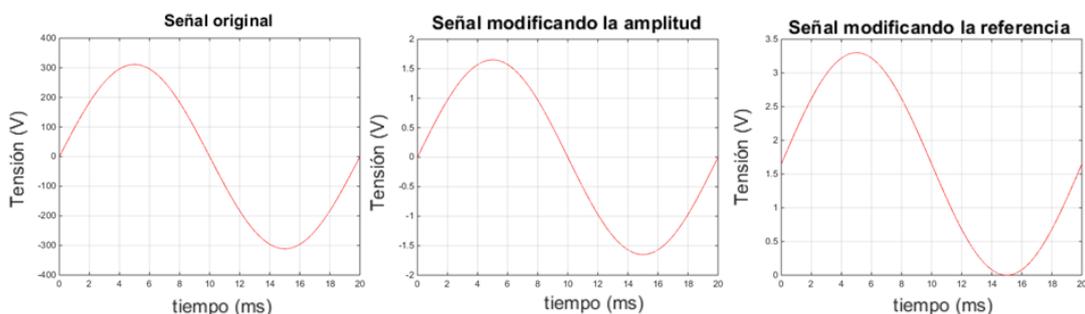


Figura 30. Ajuste de la señal de entrada a la instrumentación

Para ello, la configuración final de ambos sensores quedaría:

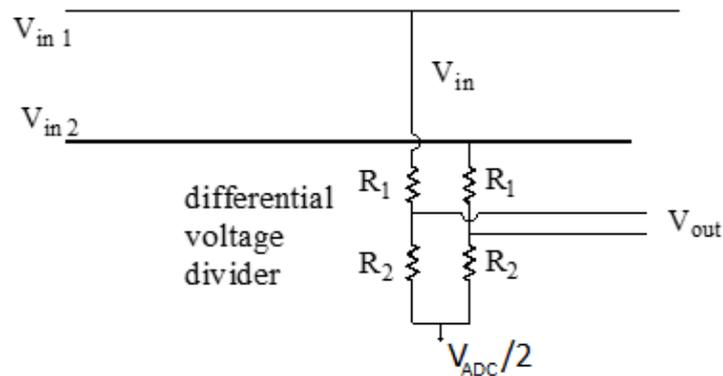


Figura 31. Divisor de tensión diferencial con referencia a un valor de tensión igual al de la alimentación del ADC medios

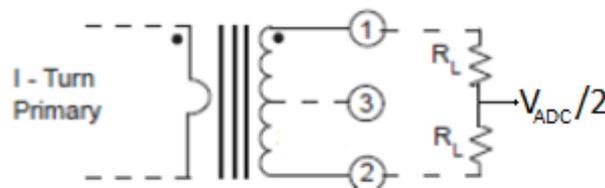


Figura 32. Transformador de corriente con referencia a un valor de tensión igual al de la alimentación del ADC medios

Nota: en el siguiente capítulo se discutirá que el hecho de modificar el valor de referencia no modifica en absoluto el valor de la tensión de salida como diferencia entre la tensión de ambos terminales.

Para fijar el valor de referencia a un valor igual a la mitad de la tensión de alimentación del ADC existen dos posibilidades relativamente simples:

- Uso de un amplificador operacional (AO). Para esta aplicación se utilizaría una configuración con realimentación negativa. El esquema sería el siguiente:

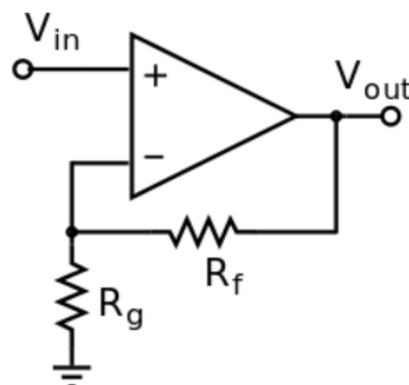
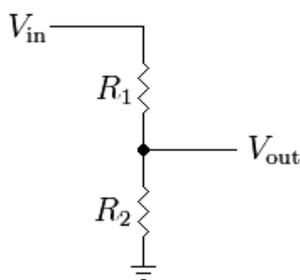


Figura 33. AO lazo cerrado

Donde la tensión  $V_{out}$ , será la tensión de referencia de los sensores de V y I, y  $V_{in}$  la tensión de alimentación del ADC. La tensión de salida, o de referencia, viene dada por la siguiente expresión:

$$V_{out} = \frac{R_g}{R_g + R_f} \cdot V_{in} \quad (\text{XII.})$$

Puesto que el objetivo es tener una tensión de referencia igual a la mitad que la tensión de alimentación, de la relación anterior tenemos que para que esto se cumpla  $R_g$  y  $R_f$  deben ser iguales. Además, debido a los amplificadores operacionales se caracterizan por tener una impedancia de entrada muy grande (idealmente infinita), la corriente que circula tanto por la entrada inversora (-), como por la no inversora (+) será prácticamente nula, por lo que no existen limitaciones en este sentido para la selección de la pareja de resistencias puesto que en cualquier caso la corriente que circulará será prácticamente cero. Notar que, para esta configuración, la tensión de entrada  $V_{in}$  y la tensión de alimentación del amplificador operacional ( $V_{cc}$ ) es la misma. Lo cual no entraña ningún problema, ya que en cualquier caso la tensión de entrada será constante, y puesto que la ganancia es 0,5 debido a que queremos tener una señal de entrada igual a la mitad que la tensión de alimentación, el amplificador operacional nunca llegará a saturar.



- Uso de un divisor de tensión. El funcionamiento es como el descrito en el sensor de tensión. En este caso, la tensión de entrada sería la de saturación del convertidor, y la tensión de salida, vendría dada por la siguiente expresión:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2} \quad (\text{XIII.})$$

Figura 34. Divisor de tensión

Así pues, la impedancia de las resistencias sería la misma para que la tensión de salida fuera la mitad. El valor de ambas resistencias debe ser tal que limite a un cierto valor la circulación de corriente hacia tierra. Asimismo, en este caso, la corriente que circula por ambas ramas del divisor de tensión no será la misma, como se discutirá en el siguiente capítulo, es por esto por lo que será necesario, además, fijar la tensión mediante dos condensadores como se muestra en la siguiente figura:

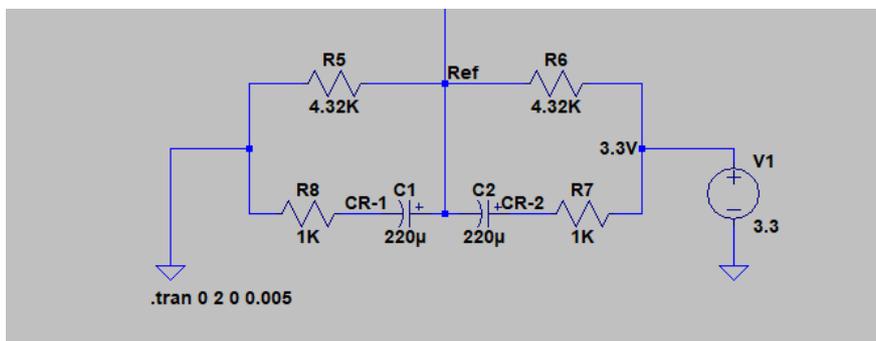


Figura 35. Esquema circuito para fijar la referencia

En el montaje anterior, notar que se deben incluir dos resistencias de arranque para limitar el flujo de corriente cuando el condensador está descargado.

De estas dos alternativas, la que finalmente se implementará es el divisor de tensión, debido a que esta solución es más sencilla y por lo tanto más robusta. Notar que la diferencia en el precio entre un montaje y otro radica exclusivamente en el uso o no de un amplificador operacional frente al uso de condensadores y resistencias, sin embargo, la vida útil de un condensador es superior al de un amplificador operacional, y siendo la diferencia en el precio entre ambos montajes marginal, se seleccionará finalmente el montaje con resistencias y condensadores para fijar la referencia.

## 4. Diseño de los sensores de tensión y corriente.

El primer punto a tratar para el diseño del equipo serán los sensores de tensión y corriente, elementos con los cuales seremos capaces de medir los valores de tensión y de corriente, con la finalidad de obtener una señal de ambas magnitudes dentro de un rango en el cual podamos leer con el convertidor analógico a digital (en adelante convertidor ADC).

Para ello, en primer lugar, se identificará qué valores tendrán ambas magnitudes.

### 4.1. Características de la red del suministro doméstico

En cuanto a las características del suministro de la energía eléctrica en uso doméstico, se tiene que la tensión de suministro es de 220 V<sub>ef</sub>. Por otro lado, la potencia máxima contratada es de 10 kW, tal y como se define en el Capítulo I del Real Decreto 216/2014, por el que se establece la metodología de cálculo de los precios voluntarios para el pequeño consumidor de energía eléctrica y su régimen jurídico de contratación<sup>(14)</sup>, con lo que la corriente máxima que se podría consumir sería de:

$$I = \frac{P}{(V \cdot \cos\phi)} = \frac{10000 \text{ W}}{220 \cdot 0,95 \text{ V}} = 47 \text{ A} \quad (\text{XIV.})$$

En el cálculo anterior se ha tomado 0,95 como valor aproximado para el factor de potencia. Así pues, el rango de la tensión será de  $\pm 220 \text{ V}_{\text{ef}}$ , mientras que, para la corriente, como máximo llegará a  $\pm 47 \text{ A}_{\text{ef}}$ .

Ya que tenemos definido el rango de las magnitudes de V y I a la entrada de los sensores, tenemos que definir qué rango queremos que tengan las señales a la salida. Esta se conoce puesto que es la tensión de alimentación del microcontrolador, el cual dará una tensión de alimentación al convertidor ADC de 3,3 V, por lo que si tenemos valores que estén fuera del rango de 0 – 3,3 V, el convertidor ADC se saturará, truncando el valor convertido al máximo/mínimo.

### 4.2. Sensores

Para el diseño de un equipo de captura y reporte de datos de consumo eléctrico, será necesario obtener por un lado valores instantáneos de tensión y corriente, para poder obtener de este modo la potencia activa instantánea y valores medios de potencia. Para la obtención de estas magnitudes, será necesario disponer de unos sensores capaces de adecuar los valores de tensión y corriente que medimos en el cuadro general a una señal que pueda ser manejable por el equipo. Esta selección de sensores ya se ha abordado en el capítulo anterior. En el presente capítulo se realizará el diseño de los mismos con la solución final adoptada.

#### 4.2.1. Diseño de los sensores

##### 4.2.1.1. Divisor de tensión diferencial

La relación entre la tensión de entrada y de salida en un divisor de tensión viene dado por:

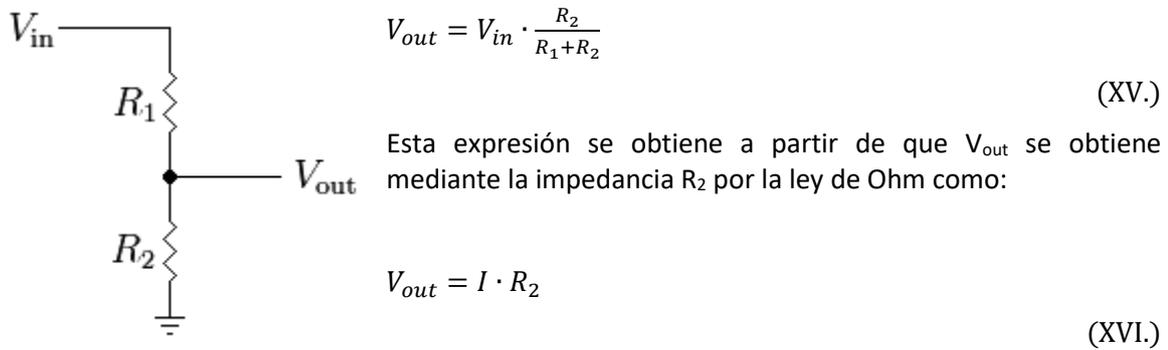


Figura 36. Divisor de tensión

Además, la corriente que circula desde  $V_{in}$  hasta tierra es, por la ley de Ohm:

$$I = \frac{V_{in}}{R_1 + R_2} \quad (XVII.)$$

Si sustituimos (XVII) en (XVI), despejando obtenemos la expresión (XV).

Notar que, para el caso de un divisor de tensión diferencial, tendríamos:

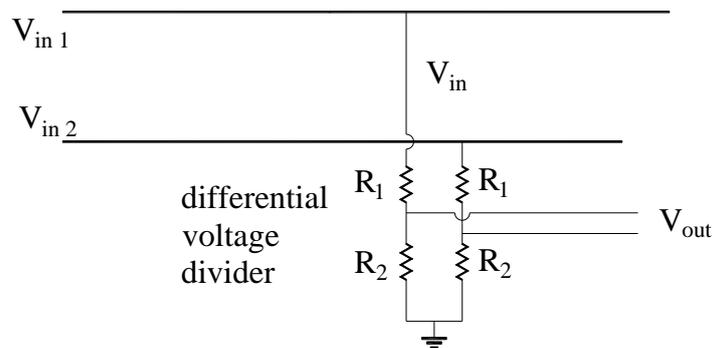


Figura 37. Divisor de tensión diferencial

$$V_{out} = V_{out1} - V_{out2} = V_{in1} \cdot \frac{R_2}{R_1 + R_2} - V_{in2} \cdot \frac{R_2}{R_1 + R_2} = (V_{in1} - V_{in2}) \cdot \frac{R_2}{R_1 + R_2} \quad (XVIII.)$$

Los criterios para el diseño del divisor de tensión diferencial son:

- Evitar que el divisor de tensión provoque interferencias con la carga
- Evitar que el divisor de tensión provoque interferencias en la instrumentación

Por otro lado, como condiciones de contorno, tenemos que la tensión entre fase-neutro es de 220 V<sub>ef</sub> y deseamos una tensión de salida con una amplitud de 3,3 V y además deseamos desplazar la tensión de forma que la onda oscile entre 0 y 3,3 V, puesto que el convertidor ADC no puede trabajar con valores negativos. Esto es posible modificando la referencia, de modo que podemos sustituir tierra por 3,3/2 V para desplazar la onda y tener siempre valores positivos.

Gráficamente, se está tratando la señal de tensión de la siguiente forma:

Partimos de la señal original, por ejemplo, la tensión, con un valor eficaz de 220 V, por lo que tenemos que la señal oscila entre  $\pm\sqrt{2} \cdot 220$  V.

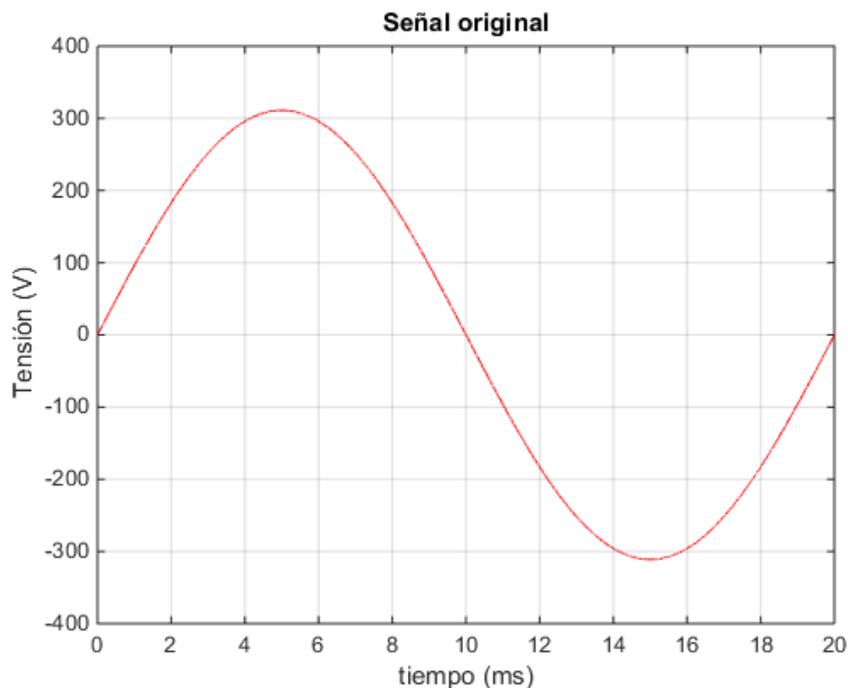


Figura 38. Forma de onda de la tensión (doméstico)

A partir de esta señal, se obtiene una proporcional a ésta con una amplitud de  $\pm \frac{3,3}{2}$  V, mediante la relación del par de resistencias que se indica en(XVIII).

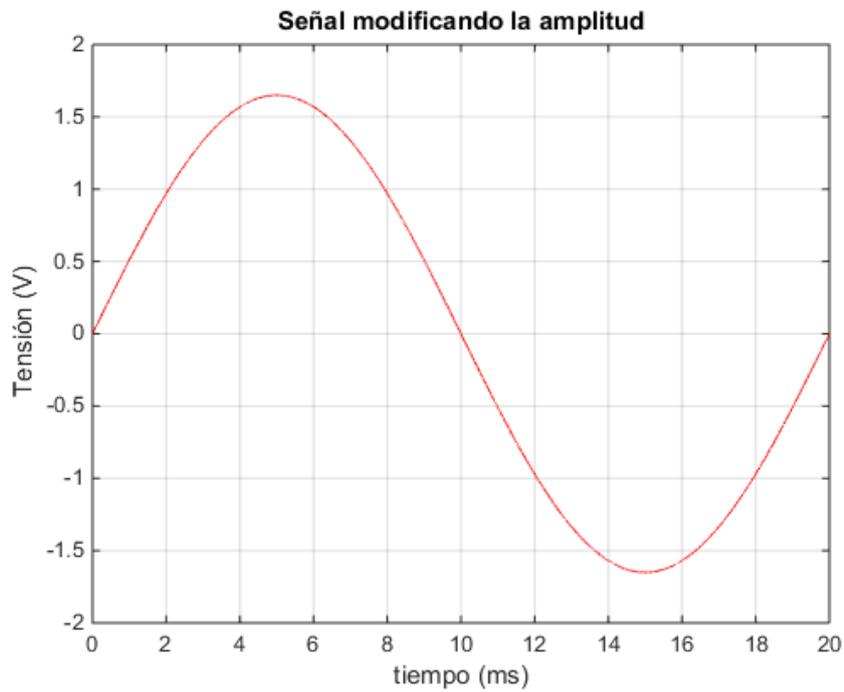


Figura 39. Forma de onda adaptando la amplitud al rango de la instrumentación

Finalmente, modificando la referencia, desplazamos la señal de forma que oscile entre 0 y 3,3 V, centrada por lo tanto en  $\frac{3,3}{2}$  V.

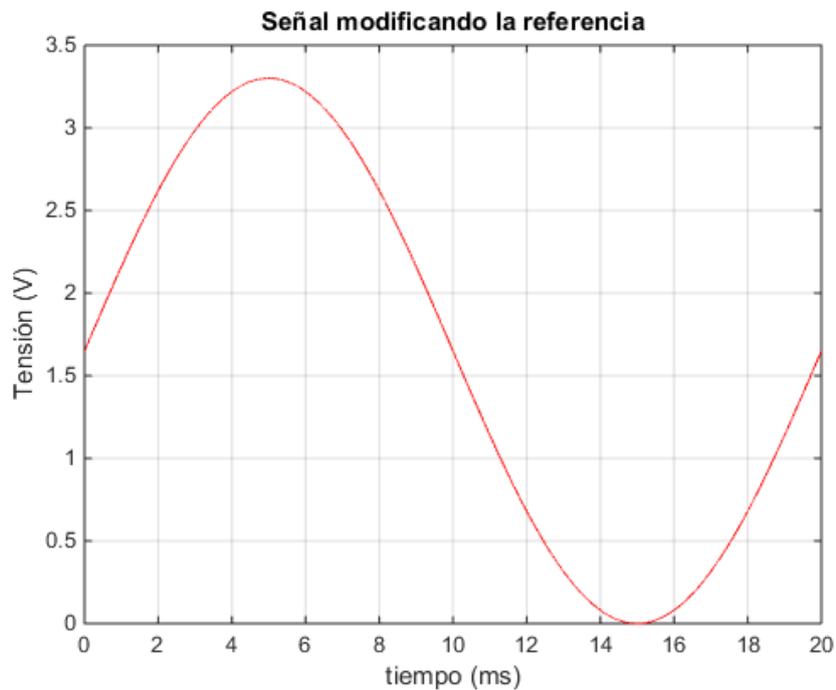


Figura 40. Forma de onda adaptando la amplitud y desplazando la onda al semieje positivo

El hecho de variar la referencia no modifica la expresión(XVIII), puesto que, si tenemos un valor de referencia distinto de cero, tendríamos que:

$$V_{out} = V_{out1} - V_{out2} = \left( V_{in1} \cdot \frac{R_2}{R_1+R_2} + V_{ref} \right) - \left( V_{in2} \cdot \frac{R_2}{R_1+R_2} + V_{ref} \right) \quad (\text{XIX.})$$

$$V_{out} = (V_{in1} - V_{in2}) \cdot \frac{R_2}{R_1+R_2} \quad (\text{XX.})$$

Así pues, sustituyendo de la expresión(XX), tenemos que las resistencias  $R_1$  y  $R_2$ , deben tener la siguiente relación:

$$V_{out} = 3,3 = V_{in} \cdot \frac{R_2}{R_1+R_2} = 2 \cdot \sqrt{2} \cdot 230 \cdot \frac{R_2}{R_1+R_2} \quad (\text{XXI.})$$

Resolviendo:

$$\frac{R_2}{R_1+R_2} = 0,005 \quad (\text{XXII.})$$

Con esto ya solo los queda una condición para tener definido el divisor de tensión. La condición que falta se puede obtener a partir de los datos del fabricante del convertidor ADC. Así pues, según el fabricante, se tiene que el convertidor cuenta con una resistencia de 1 k $\Omega$  y un condensador que toma la tensión que se muestrea cada cierto tiempo t. Gráficamente podemos ver a continuación la estructura de la etapa de entrada del convertidor:

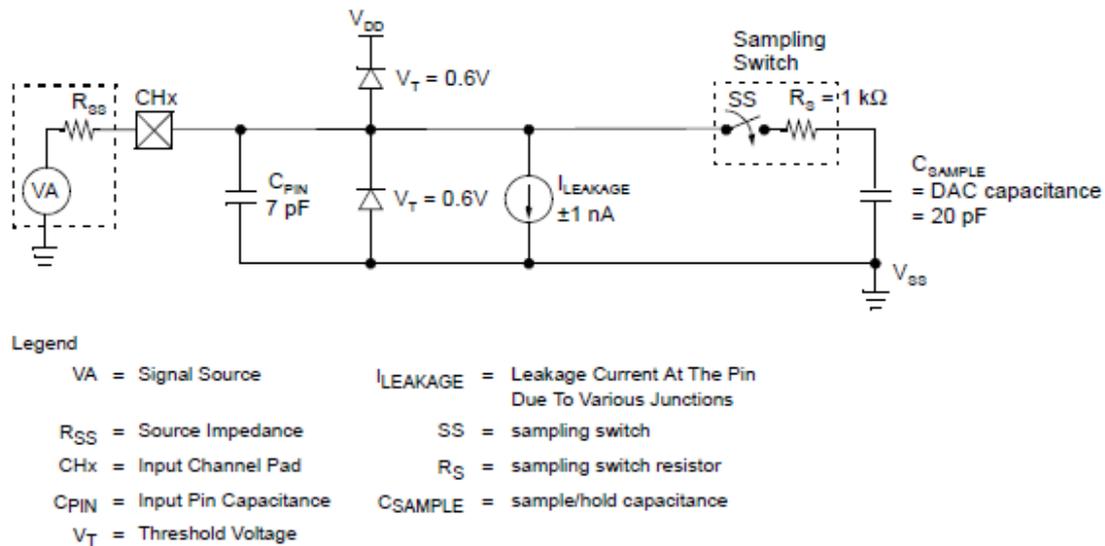


Figura 41. Modelo del convertidor ADC – Entrada analógica

De esta forma, a partir de los datos anteriores se puede estimar la  $R_{inst}$  y con ello  $R_1$  y  $R_2$ . El caso más desfavorable se presenta cuanto menor sea  $R_{inst}$ , pues también lo serán  $R_1$  y  $R_2$ , y cuanto menor sea  $R_1$  mayor será la potencia consumida por la resistencia.

Así pues, la  $R_{inst}$  equivalente más desfavorable se da en el caso con el condensador descargado. En esta situación, toda la caída de tensión se da en la resistencia  $R_s$  ( $1k\Omega$ ), por lo que  $R_{inst}$  será igual a ésta resistencia. Puesto que  $R_2$  debe ser al menos dos órdenes de magnitud menor que  $R_{inst}$  para no interferir con la instrumentación, entonces  $R_2$  será igual a  $10\Omega$ , y de (IX),  $R_1$  será en torno a  $2k\Omega$ , lo que supone un consumo de:

$$P = \frac{V^2}{R} = 26 \text{ W} \quad (\text{XXIII.})$$

Lo cual es una potencia consumida que no es asumible, por lo que hay que irse a una situación que se dé más frecuentemente con la que obtengamos una resistencia mayor. Notar que esta situación únicamente se da durante el primer muestreo una vez alimentado el circuito integrado, por lo que se podría descartar esta muestra y tomar como válidas las siguientes. En la práctica, este proceso tiene lugar durante el arranque del microcontrolador, por lo que ya ha finalizado en el momento de solicitar la primera captura.

Si pasamos a una situación más real, tenemos que, la resistencia de instrumentación será igual a:

$$R_{inst} = \frac{V_A}{I} \quad (\text{XXIV.})$$

Siendo  $V_A$  la señal que se ha capturado, y la corriente se obtendría para el proceso de carga del condensador a partir del último valor capturado:

$$I = \frac{V_{R_s}(t)}{R_s} = \frac{V_A(t) - V_C(t)}{R_s} \quad (\text{XXV.})$$

Supondremos que  $V_C(t_i)$  es igual a  $V_A(t_{i-1})$  en el instante anterior, suponiendo de esta forma que el condensador del convertidor ADC no tiene suficiente tiempo para descargarse entre el instante de tiempo  $t_{i-1}$  y  $t_i$ . La situación más desfavorable en este caso la tenemos con la máxima variación entre  $V_A(t)$  y  $V_C(t)$ . Puesto que  $V_A$  es una función senoidal, la máxima variación la tenemos cuando el seno es 0, esto se da en  $t=0$  s. Puesto que el objetivo es tomar 100 muestras en un ciclo, esto implica que el siguiente tiempo de muestreo será a los 0,0002 s. Sustituyendo tenemos las siguientes expresiones:

$$V_A(t) = \frac{3,6}{2} \cdot \text{sen}(2\pi f \cdot t) + \frac{3,6}{2} \quad (\text{XXVI.})$$

$$V_C(t) = V_A(t - 0,0002) \quad (\text{XXVII.})$$

Para  $t=0,0002$  s, tenemos que  $V_A - V_C$  vale 0,11V. Sustituyendo en (XXV), la corriente que circula es de 0,11 mA, luego la resistencia de instrumentación es igual a 16 k $\Omega$ , y  $R_2$  y  $R_1$  serán 160  $\Omega$  y 32 k $\Omega$  respectivamente. Con 32 k $\Omega$  la potencia consumida es de 1,51 W, que sigue siendo una potencia que no se puede asumir.

Haciendo un análisis más detallado, tenemos que la impedancia de salida del divisor de tensión es:

$$R_o = \frac{R_2}{2} \quad (\text{XXVIII.})$$

Teniendo lo siguiente:

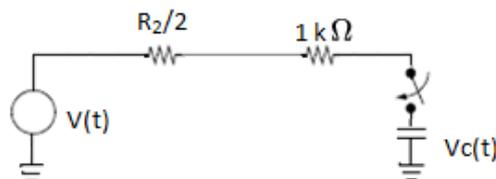


Figura 42. Esquema señal analógica con modelo simplificado del convertidor ADC

Donde  $V(t)$  es la señal de tensión a la salida del divisor de tensión y  $V_c(t)$  es la tensión del condensador. Tenemos como restricción para el valor de  $R_2$  que debemos ser capaces de tener una tensión en el condensador igual a  $V(t_i)$  en  $t=t_i$ , partiendo de una tensión  $V(t_{i-1})$  en el condensador en  $t=t_{i-1}$ . El tiempo que tiene para cargarse a la tensión  $V(t_i)$  entre  $t_{i-1}$  es igual al tiempo que tarda desde el flanco de subida del quinto pulso de reloj y hasta el flanco de bajada del siguiente pulso de reloj (ver 5.3.1 Comunicación SPI), siendo este tiempo aproximadamente igual a  $1,9 \mu s$ .

Así pues, tenemos de esta forma un circuito RC en serie en el cual el condensador se carga. La expresión que define el comportamiento durante el transitorio sigue la siguiente expresión:

$$V(t) = V_R(t) + V_c(t) \quad (\text{XXIX.})$$

$$V(t) = R \cdot i(t) + \frac{1}{C} \int_{t_0}^{t_1} i(t) dt \quad (\text{XXX.})$$

Donde se agrupa en  $R$  tanto  $R_0$  como  $R_s$ .

Resolviendo por el método clásico del operador  $D$  ( $D = \frac{d}{dt}$ ), tenemos lo siguiente:

$$DV(t) = R \cdot Di(t) + \frac{1}{C} i(t) \quad (\text{XXXI.})$$

Notar que la tensión  $V(t)$ , es la tensión que se muestrea, la cual se puede asumir constante entre una muestra y otra, por lo que  $\frac{d}{dt} V(t) = 0$ , teniendo lo siguiente:

$$0 = R \cdot Di(t) + \frac{1}{C} i(t) \quad (\text{XXXII.})$$

Siendo la siguiente función solución a la ecuación característica anterior:

$$i(t) = K_1 e^{-\frac{1}{RC}t} \quad (\text{XXXIII.})$$

Para obtener  $K_1$ , tenemos como condición inicial que:

$$i(0) = \frac{V - V_{C,0}}{R} \quad (\text{XXXIV.})$$

Sustituyendo en (XXXIII.):

$$i(t) = K_1 e^{-\frac{1}{RC}t} = \frac{V - V_{C,0}}{R} e^{-\frac{1}{RC}t} \quad (\text{XXXV.})$$

De esta forma, la función temporal de la tensión del condensador es:

$$V_C(t) = \frac{1}{C} \int_{t_0}^{t_1} i(t) dt = \frac{1}{C} \int_{t_0}^{t_1} \left( \frac{V - V_{C,0}}{R} e^{-\frac{1}{RC}t} \right) dt = (V - V_{C,0}) \cdot \left( 1 - e^{-\frac{1}{RC}t} \right) + K_2 \quad (\text{XXXVI.})$$

Donde, para obtener  $K_2$ , tenemos como condición inicial que  $V_C(t = 0)$  tendrá el valor de la última muestra tomada, esto es igual a  $V(t = -0,0002)$  y será por lo tanto un valor constante, por lo que:

$$V_{C,0} = (V - V_{C,0}) \cdot \left( 1 - e^{-\frac{1}{RC}t} \right) + K_2 = (V - V_{C,0}) \cdot (1 - e^0) + K_2 \quad (\text{XXXVII.})$$

$$K_2 = V_{C,0} \quad (\text{XXXVIII.})$$

Teniendo finalmente que:

$$V_C(t) = \frac{1}{C} \int_{t_0}^{t_1} i(t) dt = \frac{1}{C} \int_{t_0}^{t_1} \left( \frac{V - V_{C,0}}{R} e^{-\frac{1}{RC}t} \right) dt = (V - V_{C,0}) \cdot \left( 1 - e^{-\frac{1}{RC}t} \right) + V_{C,0} \quad (\text{XXXIX.})$$

Gráficamente, tenemos que la evolución temporal de la tensión del condensador es:

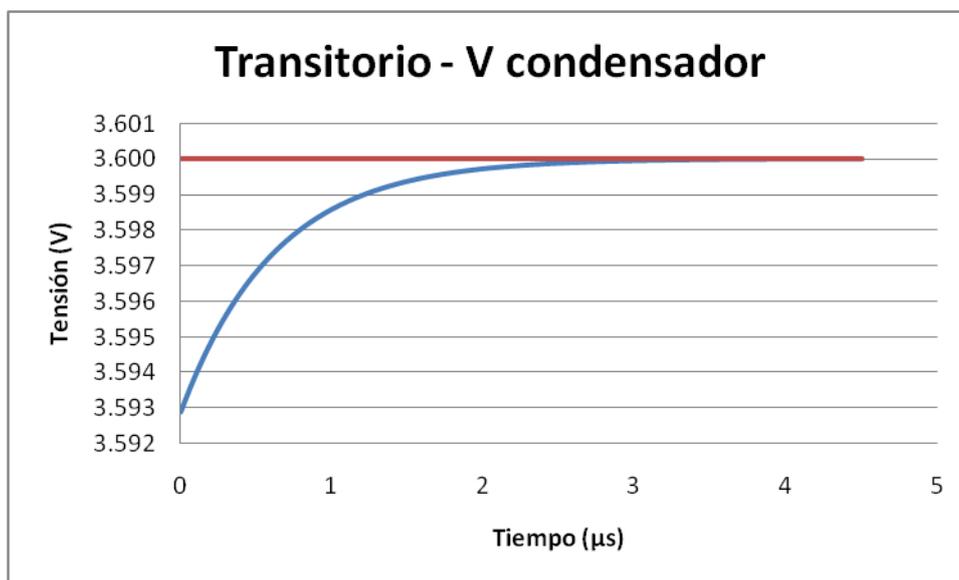


Figura 43. Evolución temporal de un condensador cargándose

Para un valor de R y C dado tenemos que la tensión del condensador durante la carga evoluciona siguiendo una exponencial negativa. Cuanto mayor sea el producto de R y C, más tardará en cargarse, y cuanto menor sea, se cargará más rápidamente. A este producto se le conoce como la constante de tiempo, y se representa mediante la letra  $t$ . Como se puede observar, el proceso de carga de condensador es infinitamente largo, sin embargo, en la práctica, se sabe que la diferencia entre la tensión actual y la de régimen permanente es del orden del 0,7% (precisión aceptable) cuando el tiempo valga 5 veces la constante de tiempo. Esto es:

$$t = 5 \cdot RC \tag{XL.}$$

Despejando:

$$\frac{t}{5} = R \cdot C \tag{XLI.}$$

Sustituyendo:

$$\frac{t}{5} = \left(\frac{R_2}{2} + R_s\right) \cdot C \tag{XLII.}$$

Finalmente tenemos:

$$R_2 = \left( \frac{t}{5 \cdot C} - R_s \right) \cdot 2 = 18 \text{ k}\Omega \quad (\text{XLIII.})$$

Donde t vale 1,9  $\mu\text{s}$ , puesto que es el tiempo de carga que tendremos para llegar a la tensión como se indicará en 5.3.1 *Comunicación SPI* de la fuente,  $R_s$  vale 1K $\Omega$ , y C vale 20 pF (especificaciones del fabricante).

Luego de (XXVIII.)  $R_1$  es igual a 3,6 M $\Omega$  y la potencia consumida será de 13,4 mW. Esta pareja de valores serían los máximos aceptables desde el punto de vista de la impedancia de entrada de convertidor AD/DC.

Desde el punto de vista de la potencia consumida por las resistencias del divisor de tensión, teniendo como objetivo utilizar resistencias de coste reducido con una potencia nominal de 0,25W, el valor más reducido de las resistencias de la rama superior del divisor es:

$$R_1 \leq (U_{red}^2 / P_{res}) = 193,6 \text{ k}\Omega \quad (\text{XLIV.})$$

Este valor es inferior al que marca el diseño basado en la etapa de entrada del ADC y, por lo tanto, válido desde ese punto de vista. Para evitar forzar las resistencias de la rama superior a su potencia nominal se escoge finalmente un valor de 301k $\Omega$  para  $R_1$  y de 1,3k $\Omega$  para  $R_2$ .

#### 4.2.1.2. Transformador de corriente

Para la selección del transformador de corriente, se tiene que el valor de la corriente de entrada será teniendo en cuenta que la potencia máxima contratada es de 10 kW y tomando como referencia un factor de potencia de 0,95:

$$P = V \cdot I \cdot \cos(\varphi) \quad (\text{XLV.})$$

$$I = \frac{P}{V \cdot \cos(\varphi)} = \frac{10 \cdot 10^3 \text{ W}}{230 \cdot 0,95 \text{ V}} = 45 \text{ A} \quad (\text{XLVI.})$$

Al igual que en el caso anterior, tenemos que la señal debe ser positiva. Para esto utilizaremos un divisor de tensión que nos fije una tensión de 3,3/2 V entre las resistencias que unen los terminales 1 y 2. Notar que en este caso podemos hacer uso de un divisor de tensión para fijar la referencia puesto que la corriente no se puede cerrar por otro camino que no sea de 1 a 2. Gráficamente lo podemos ver a continuación:

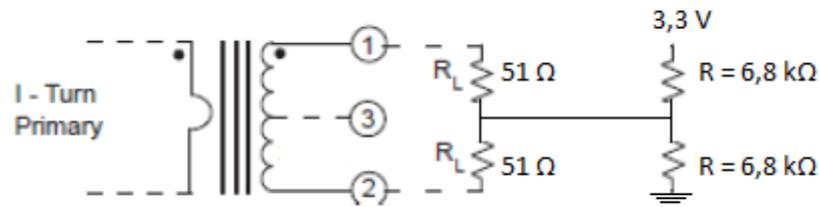


Figura 44. Esquema transformador de corriente

Así pues, a la entrada tenemos una señal senoidal cuyo valor máximo es de 45 A<sub>ef</sub> y deseamos tener a la salida una señal que oscile en 3,3/2 V, y que no sea menor de 0V ni mayor de 3,3V. Con esto tenemos que la relación de espiras debe ser:

$$\frac{I_1}{I_2} = \frac{N_2}{N_1} = \frac{1}{r_t} = \frac{2 \cdot \sqrt{2} \cdot 45}{\frac{3,3}{102}} = 3934$$

(XLVII.)

Luego la relación debe ser aproximadamente de 4000:1.

#### 4.2.1.3. Fijar la referencia

Como ya se ha indicado, para fijar la referencia, se hará uso de un divisor de tensión, con un valor para las resistencias tal que limiten el flujo de corriente teniendo en cuenta la tensión de 3,3 V.

Sin embargo, aunque con el transformador de corriente tenemos que ambos circuitos son independientes (puesto que la corriente que circula por el transformador sólo se puede cerrar por ese camino: el propio transformador de corriente) para el caso del sensor de tensión la corriente sí que se podría cerrar por la resistencia que forma el divisor de tensión que fija la referencia, lo que modificaría el valor de tensión que deseamos. Por este motivo, será necesario hacer uso de condensadores en paralelo que fijen la tensión de la referencia. La estructura del circuito de referencia de tensión sería:

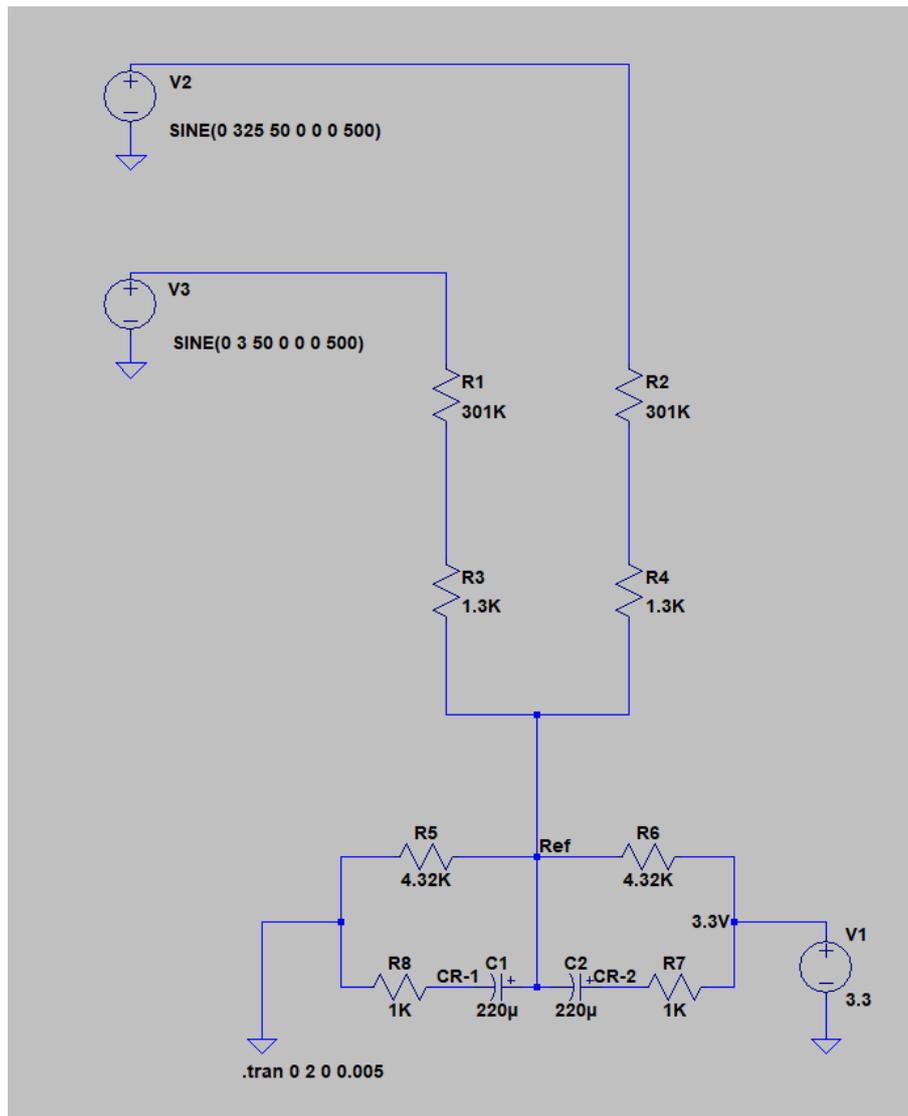


Figura 45. Esquema montaje final divisor de tensión

Donde el valor de los condensadores se obtiene mediante simulación, para limitar el rizado de la señal de referencia. El circuito de referencia utiliza dos resistencias de carga para limitar la corriente de carga mientras que los condensadores se están cargando y que no se produzca un cortocircuito en la fuente de 3,3V.

Realizando simulaciones, tenemos que con un valor para los condensadores de 220  $\mu\text{F}$  el rizado es de  $\pm 0,5$  V. Notar que, finalmente se ha optado por un valor para las resistencias del divisor de tensión que nos da la señal para el convertidor de 301 K $\Omega$  y 1,3 K $\Omega$  que resultan en un consumo con el diseño planteado en la figura anterior menor de  $\frac{1}{4}$  W.

Asimismo, podemos ver que la corriente que circula por las resistencias que forman el divisor de tensión que fija la referencia es casi 0 una vez cargado, lo que asegura que no estamos interfiriendo para la lectura con el divisor de tensión. Esto lo vemos gráficamente en las siguientes figuras:

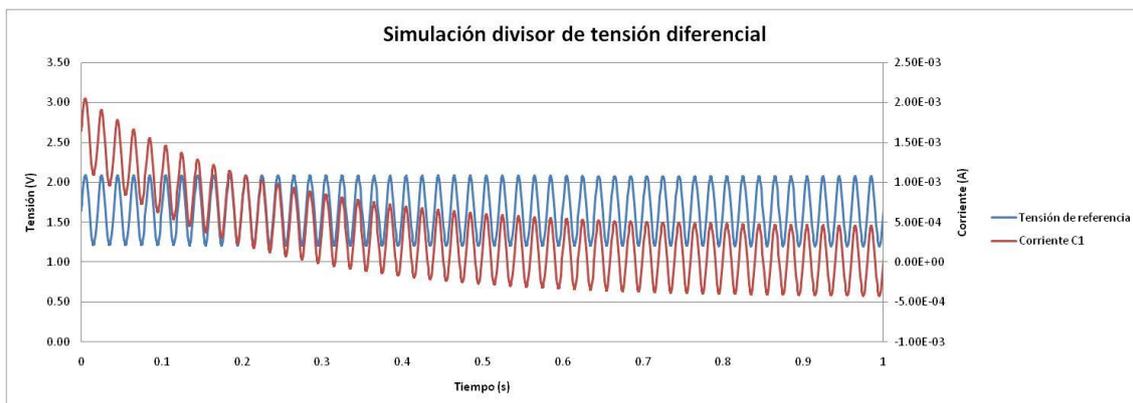


Figura 46. Rizado de la tensión de referencia y consumo del condensador C1 que fija la referencia

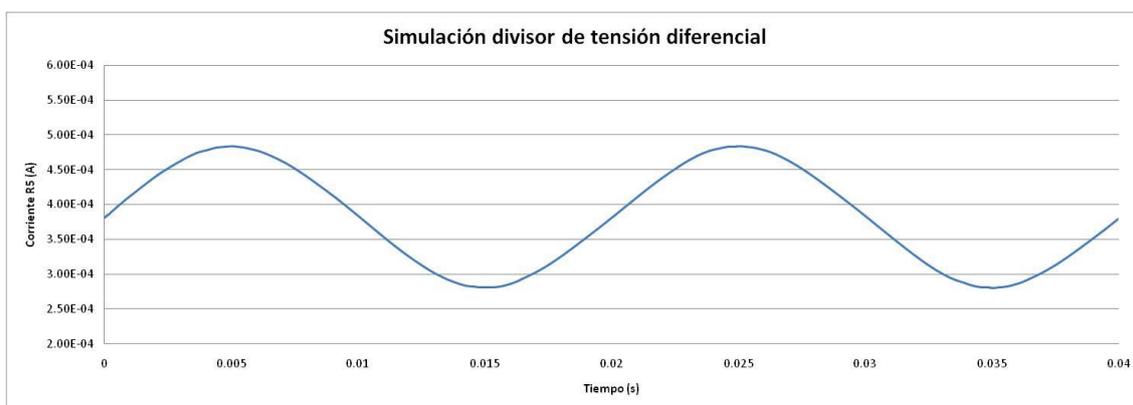


Figura 47. Consumo del divisor de tensión que fija la referencia

Por otro lado, el rizado que medimos en la simulación, resulta asumible para la corriente, puesto que en pocas ocasiones estaremos cerca del máximo ( $45 A_{ef}$ ), pero para la tensión, tenemos gráficamente que la diferencia de potencial respecto a tierra de los dos terminales del divisor de tensión son:

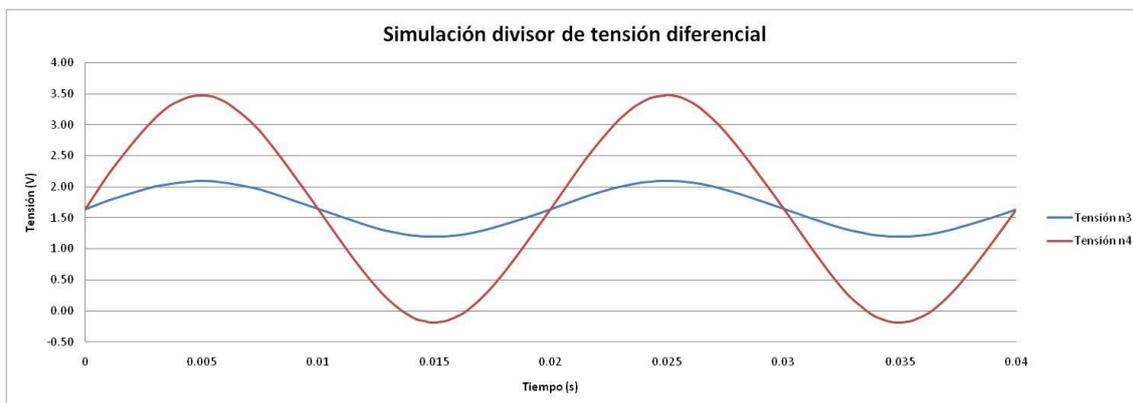


Figura 48. Señales de tensión obtenidas - n3 entre R1 y R3; n4 entre R2 y R4

La diferencia entre ambos terminales vemos que está muy cerca de la saturación, por lo que será necesario comprobar experimentalmente que no existe saturación en la lectura de la señal de tensión. Así pues, a continuación tenemos gráficamente las medidas de tensión realizadas utilizando el ADC a lo largo de un período:

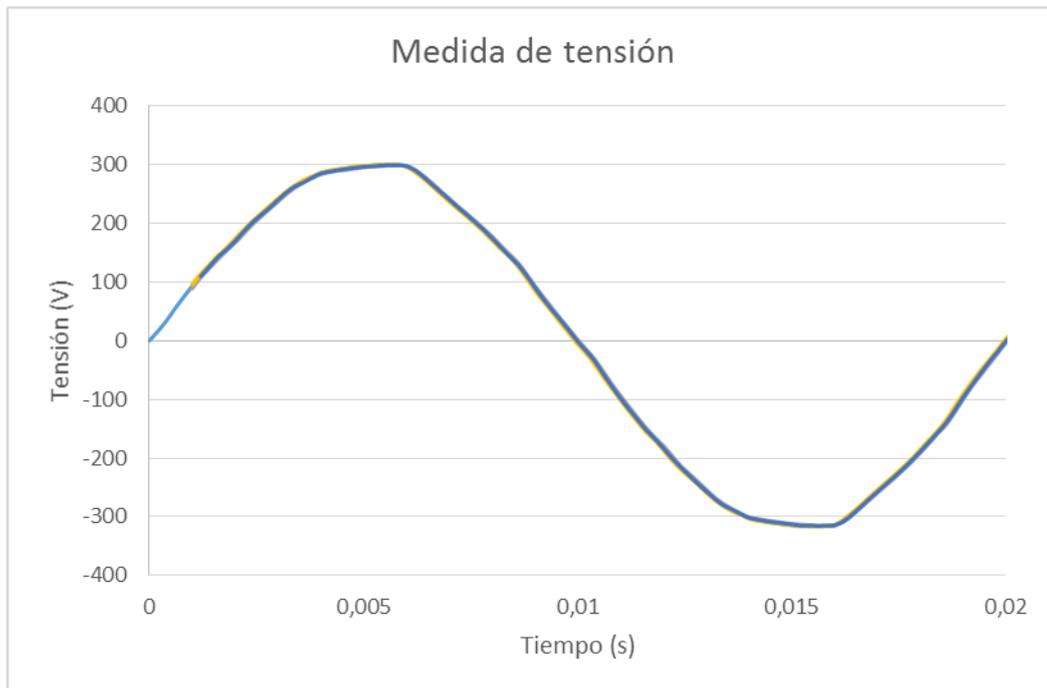


Figura 49. Forma de onda experimental obtenida de la tensión (red)

Donde vemos que no existe saturación en la conversión ADC (el truncado de la onda de tensión es el característico de redes de alimentación en las que predominan los consumos de tipo electrónico, con mayor drenaje de corriente en los intervalos de cada semiciclo en que la tensión es máxima) puesto que no se aprecian límites planos ni en el semieje positivo ni en el negativo. Estas muestras, como se explicará en capítulos posteriores, se han tomado con un tiempo de muestreo de 200  $\mu$ s, teniendo de esta forma 100 muestras por periodo.

## 5. Captura digital.

En cuanto a la instrumentación, se seleccionará como convertidor ADC, el MCP3304, sin embargo, en una fase inicial, se realizarán pruebas con el Convertidor MCP3008, y como microcontrolador el ESP8266. El motivo por el cual se seleccionan estos se detalla en los siguientes apartados.

### 5.1. Microcontrolador

Se ha seleccionado el ESP8266 debido a su bajo coste, puesto que su precio ronda los 2-3 €, además de incorporar todas las funcionalidades necesarias para la realización del proyecto, que son:

- Incorpora una antena Wifi y dispone de librerías para comunicación por estas redes, lo que nos permitirá conectarnos a un servidor para poder obtener y publicar mensajes.
- Nivel de consumo bajo, un requisito indispensable para esta aplicación es su consumo, pues éste debe ser lo más bajo posible de modo que tenga un peso residual en el consumo global del usuario.
- Dispone de una memoria flash externa que nos permite almacenar históricos, así como de diversos pines que proporciona flexibilidad y la hora de configurar el montaje. En concreto, tiene pines definidos para establecer una conexión Serial Peripheral Interface (en adelante SPI) vía hardware con el convertidor ADC.
- El microcontrolador se conectará vía USB y su tensión de operación es de 3-3,6 V, siendo capaz de alimentar a dicha tensión en pines ya definidos.

Además de las funcionalidades mencionadas anteriormente, cabe destacar que existen referencias, bibliografías y experiencias sobre el uso del ESP8266 y su conexión vía SPI, lo que permite tener una base y referencia para poder realizar el proyecto. Gráficamente, en la siguiente figura<sup>(15)(16)</sup>, se representa los pines que dispone el ESP8266, destacando los que se utilizarán para la conexión SPI vía hardware con el convertidor.

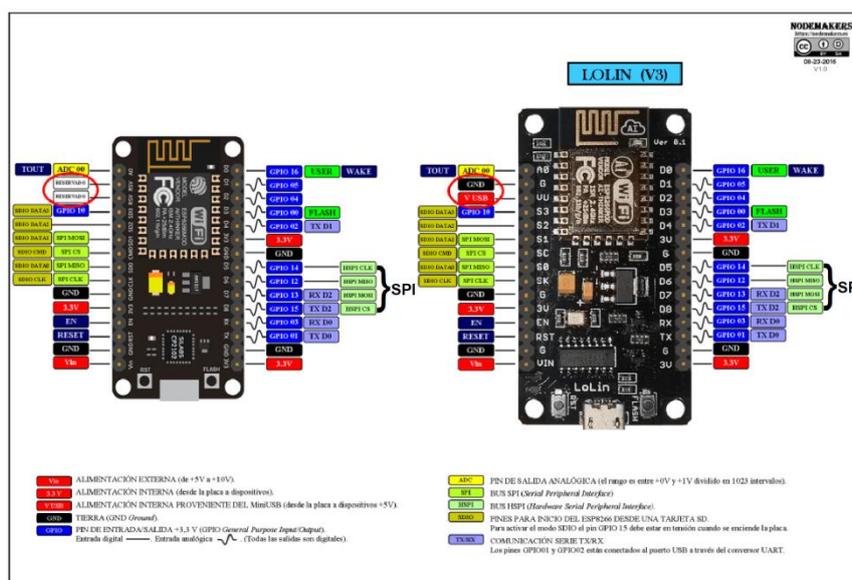


Figura 50. PINOUT-NodeMCU\_1.0-V2-y-V3

## 5.2. Convertidor ADC

### 5.2.1. MCP3008

Como se ha introducido al comienzo de este capítulo, en un primer lugar, se ha seleccionado el MCP3008, cuya configuración de pines se representa a continuación<sup>(17)</sup>.

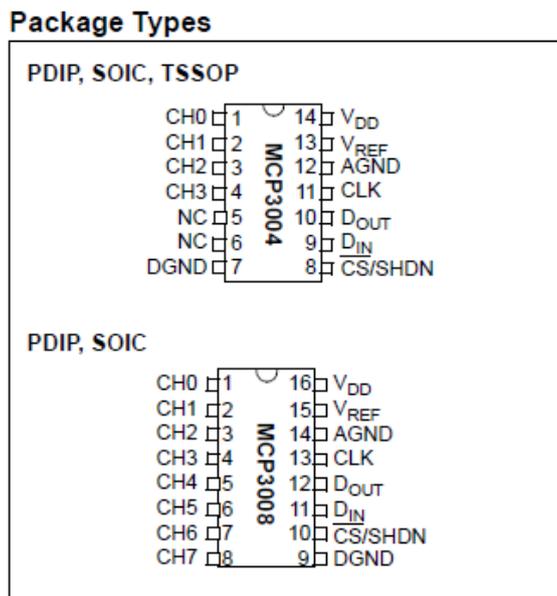


Figura 51. Configuración pines MCP3008

El motivo, es debido a que existen ejemplos y referencias<sup>(18)(19)(20)</sup> donde se establece comunicación entre el ESP8266 y el MCP3008, pudiendo de esta forma descargar las librerías de estos ejemplos, para así incorporarlas al proyecto y de esta forma realizar las primeras pruebas, más adelante hablaremos de las restricciones del MCP3008 y de los cambios necesarios para implementar la conversión AD con el MCP3304.

#### 5.2.1.1. Especificaciones

Como características principales, destacan<sup>(17)</sup>:

- Una resolución de 10-bit, lo que proporciona  $2^{10} - 1$  niveles, esto es 1023, teniendo en cuenta que la alimentación es a 3,3 V, y que se corresponde con el mayor valor de tensión, esto nos permite distinguir.

$$\Delta V_{\text{señal}} = \frac{V_{\text{max}}}{2^{10}-1} = 0,00352V$$

(XLVIII.)

Lo que se corresponde con un valor de tensión de:

$$\Delta V_{\text{tensión}} = \frac{\Delta V_{\text{señal}} \cdot \sqrt{2} \cdot 230}{3,3} = 0,343V$$

(XLIX.)

Y para la corriente, tenemos que podemos distinguir entre:

$$\Delta V_{\text{corriente}} = \frac{V_{\text{max}}}{2^{10}-1} \cdot \frac{45}{3,3} = 0,044A$$

(L.)

- Se puede programar para que la entrada sea single-ended o pseudo-diferencial. Ésta última será la utilizada finalmente, como se justificará más adelante.
- Destacar que no tiene bit de signo, por lo que en el caso de leer una entrada pseudo-diferencial, si el canal + es menor que el -, el resultado será 0.

#### 5.2.1.2. Transmisión de datos

La transmisión de datos en la comunicación SPI se da de la siguiente manera (ver capítulo 2.3.1 *Convertidores con salida en paralelo*). Gráficamente, podemos ver a continuación el orden con el que se envía la información:

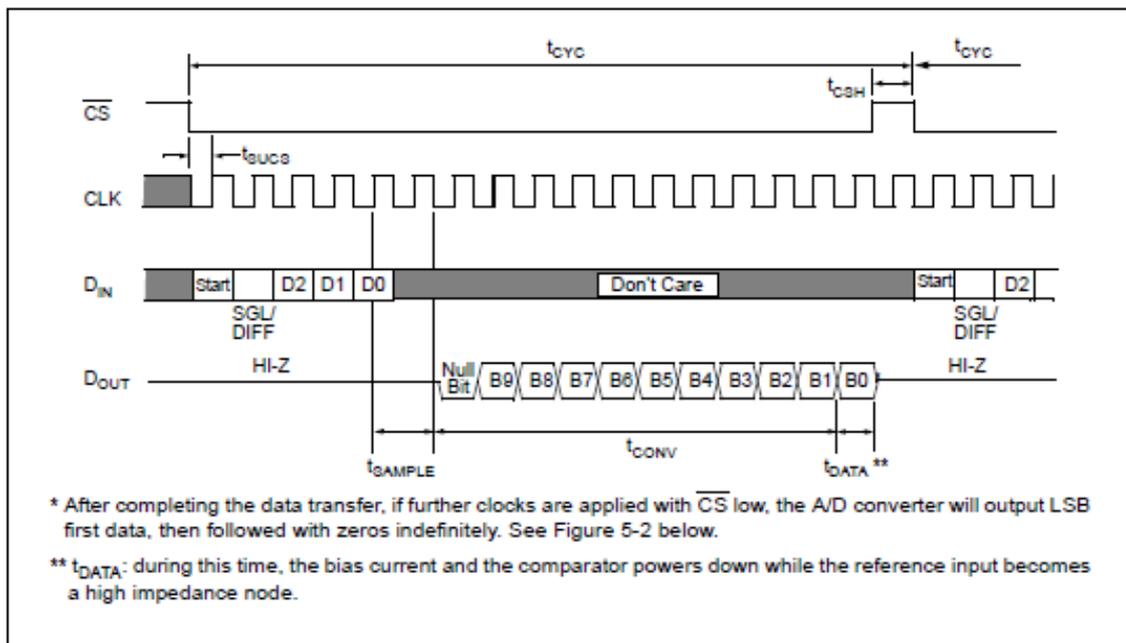


Figura 52. Comunicación SPI

Tenemos 4 señales, CS, CLK,  $D_{IN}$  y  $D_{OUT}$ , siendo estas dos última, las señales del microcontrolador y del convertidor, respectivamente.

Por defecto, el valor de la señal CS, esta alto, para iniciar la comunicación se tiene que poner a valor bajo. A continuación, el primer pulso de la señal CLK con CS bajo y  $D_{IN}$  alto constituirá el bit de inicio. El bit posterior a éste indica si la entrada es single-ended o pseudo-diferencial. Tras indicar el tipo de entrada, se especifica el canal, con los bits D0, D1 y D2, tal y como se indica en la siguiente tabla:

Control Bit Selections				Input Configuration	Channel Selection
Single /Diff	D2	D1	D0		
1	0	0	0	single-ended	CH0
1	0	0	1	single-ended	CH1
1	0	1	0	single-ended	CH2
1	0	1	1	single-ended	CH3
1	1	0	0	single-ended	CH4
1	1	0	1	single-ended	CH5
1	1	1	0	single-ended	CH6
1	1	1	1	single-ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

Figura 53. Codificación en la selección de canal entrada

A continuación, el convertidor comienza a tomar una muestra en el cuarto flanco de subida de la señal CLK, después de que se haya recibido el bit de inicio, pues es justo en este instante en el cual se conoce el tipo de entrada y el canal en el que se recibe. El periodo de muestreo termina en el flanco de bajada del quinto pulso de la señal CLK, después de que se haya recibido el bit de inicio.

De esta forma, una vez que el bit D0 es recibido, es necesario un pulso de la señal CLK para completar el periodo de muestreo. A partir de ese momento, lo que se envíe por  $D_{IN}$  es

información no relevante y el ADC no la utilizará. A continuación, en la señal  $D_{OUT}$  antes del envío del valor muestreado, se tiene un bit en valor bajo, tras el cual tenemos que para los siguiente 10 pulsos de CLK, se envía el valor muestreado. Tras terminar el bit B0, la señal CS se pone el valor alto.

### 5.2.2. MCP3304

Para este proyecto, finalmente, se ha hecho uso del MCP3304, debido principalmente a que el MCP3008, en el caso de tener una entrada pseudo-diferencial, el valor del canal + no puede ser menor que el del canal -, y será con este tipo de entradas con la que se trabajará debido al tipo de sensores que estamos manejando. Ya que la señal que se debe muestrear es la diferencia entre dos terminales, tanto en el divisor de tensión diferencial, como en el transformador de corriente:

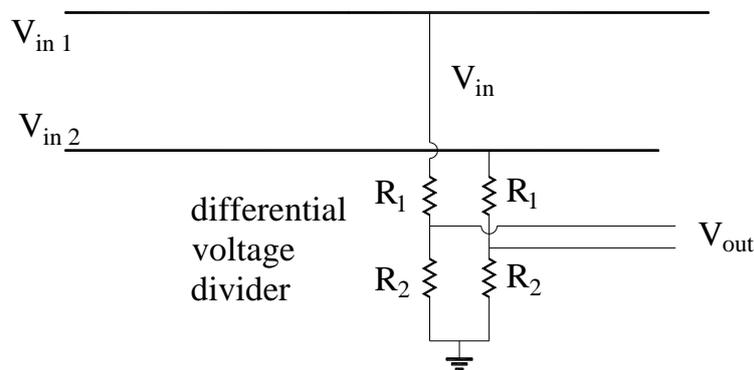


Figura 54. Divisor de tensión diferencial

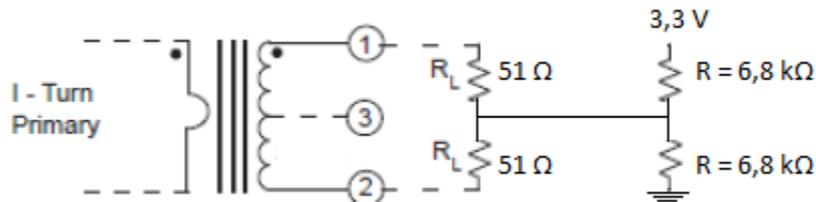


Figura 55. Esquema transformador de corriente

Teniendo en el divisor de tensión diferencial como los dos terminales, los dos nodos entre  $R_1$  y  $R_2$  de ambas ramas, y en el transformador de corriente se tiene como los dos terminales el punto 1 y 2.

Así pues, como se pudo ver en el capítulo 4.2 *Sensores*, tenemos que en cada periodo nos encontraremos con que la señal + pueda ser menor que la -, gráficamente se puede apreciar en esta simulación:

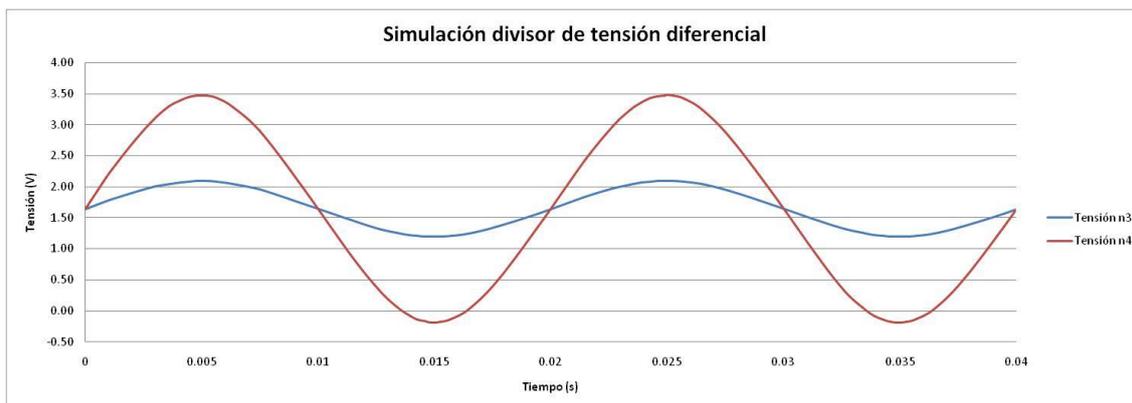


Figura 56. Señales de tensión obtenidas - n3 entre R1 y R3; n4 entre R2 y R4

Poner figuras con fondo claro para que se puedan 'leer'.

En la figura anterior se está representado cada una de las señales del divisor de tensión diferencial y, como se puede observar, la señal que se corresponde con la tensión de fase, durante el primer semi-periodo es mayor que la de neutro, pero en el segundo semi-periodo, la señal de fase es menor que la de neutro.

Es por esto, por lo que el MCP3008 resulta inviable para esta aplicación, por lo que el convertidor seleccionado finalmente será el MCP3304, cuya configuración de pines se representa a continuación:

#### Package Types

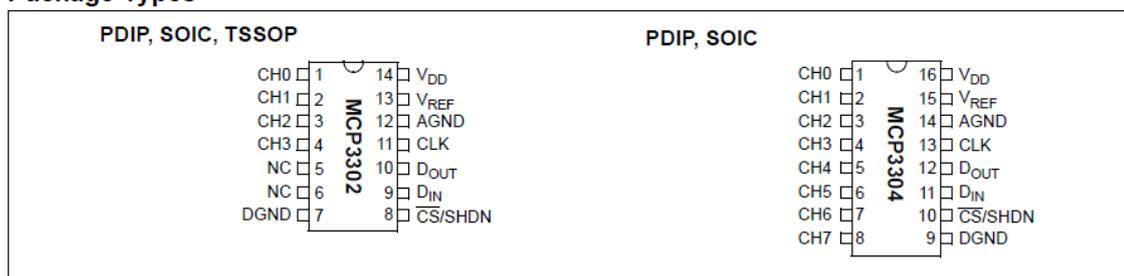


Figura 57. Configuración pines MCP3304

#### 5.2.2.1. Especificaciones

Como características principales, destacan<sup>(17)</sup>:

- Una resolución de 13-bit, teniendo en cuenta que el primer bit es de signo, por lo que tenemos  $2^{12} - 1$  niveles, esto es 4095 en positivo y negativo, es decir, desde -4096 hasta 4095 en modo pseudo-diferencial. En la siguiente figura se muestra la cuantización correspondiente al modo pseudo-diferencial, en el que se convierten a digital las señales presentes en dos entradas consecutivas (por ejemplo 0 y 1) y se transmite la diferencia entre ambas, con su signo:

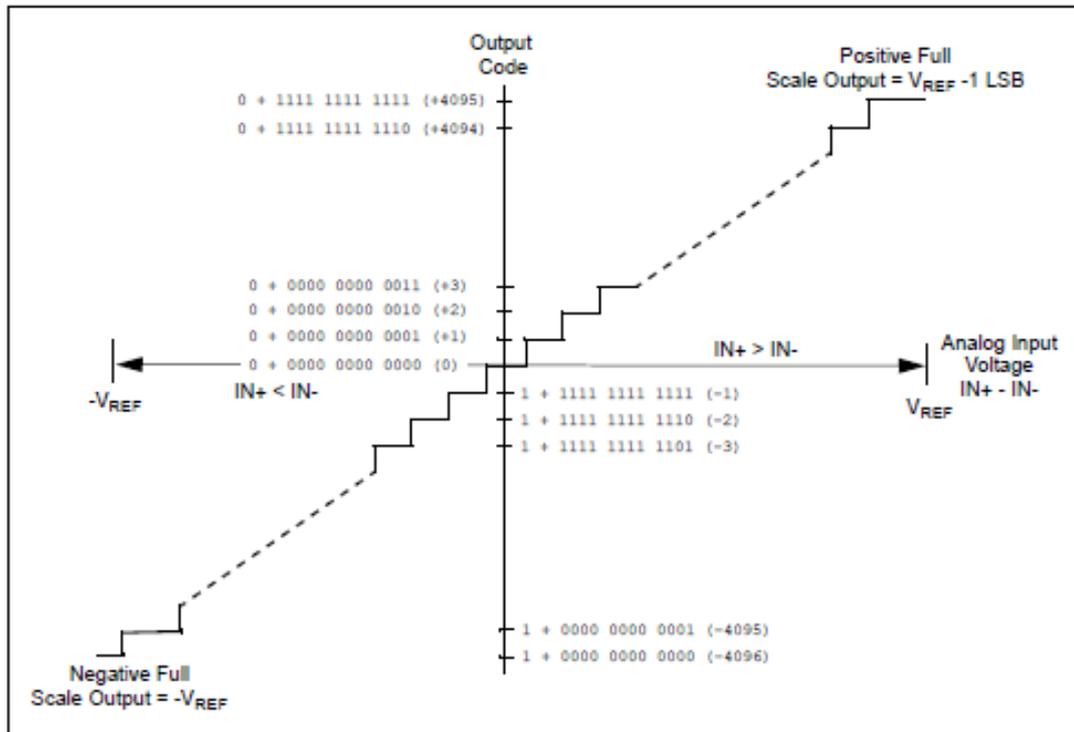


Figura 58. Función de transformación de la salida

Teniendo en cuenta que la alimentación es a 3,3 V, y que se corresponde con el mayor valor de tensión, esto nos permite distinguir.

$$\Delta V_{\text{señal}} = \frac{V_{\text{max}}}{2 \cdot (2^{12} - 1)} = 0,0004V \quad (\text{LI.})$$

Lo que se corresponde con un valor de tensión de:

$$\Delta V_{\text{tensión}} = \frac{\Delta V_{\text{señal}} \cdot \sqrt{2} \cdot 230}{3,3} = 0,04V \quad (\text{LII.})$$

Y para la corriente, tenemos que podemos distinguir entre:

$$\Delta V_{\text{corriente}} = \frac{V_{\text{max}}}{2 \cdot (2^{12} - 1)} \cdot \frac{45}{3,3} = 0,005A \quad (\text{LIII.})$$

- Se puede programar para que la entrada sea single-ended o pseudo-diferencial. Ésta última será la utilizada.

### 5.2.2.2. Transmisión de datos

La transmisión de datos es igual que con el MCP3008, salvo que en este caso, en la señal  $D_{OUT}$  tenemos que al tener una resolución de 12, sin tener en cuenta el bit de signo, contamos con los bits del B0 al B11, mientras que con el MCP3008 iban del B0 al B9. Además, antes del bit B11, tenemos que se envía en bit de signo. Gráficamente se puede ver a continuación:

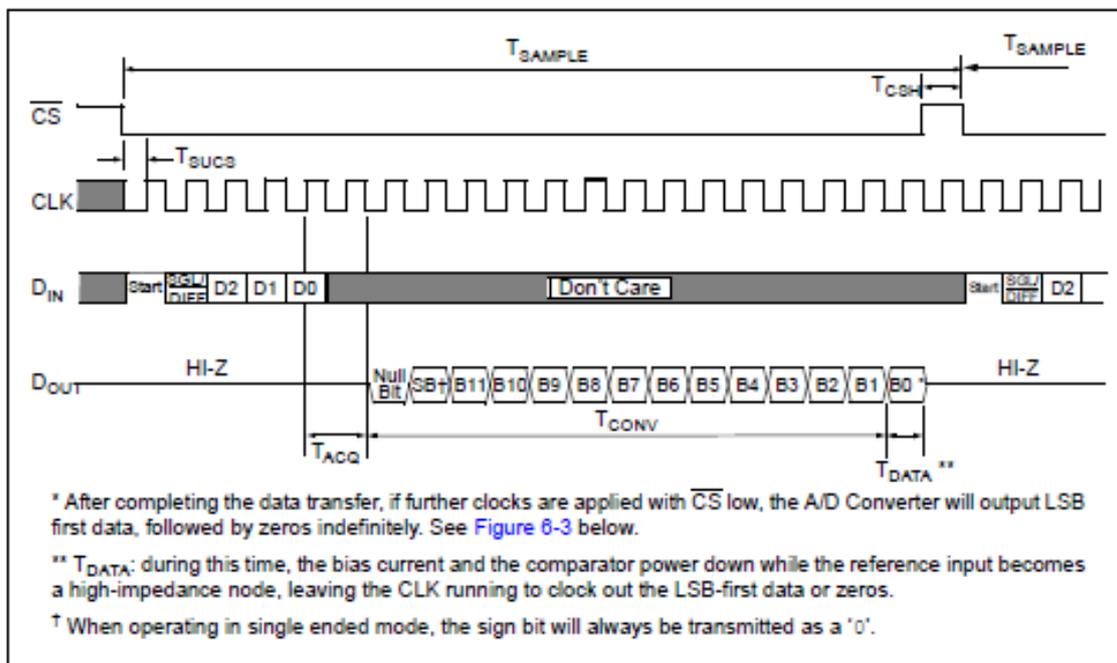


Figura 59. Comunicación SPI

## 5.3. Comunicación convertidor ADC - microcontrolador

Como ya se vio en el Capítulo 2, los tres métodos de comunicación disponibles en el ESP8266 son el I<sup>2</sup>C bus, el serial UART bus, y el SPI bus. Este último tipo de comunicación emplea líneas adicionales para enviar y recibir datos, sin embargo proporciona una mayor velocidad a la hora de enviar información. A lo largo de este punto, se describirá en profundidad cómo funciona este tipo de comunicación, además de realizar pruebas para verificar su correcto funcionamiento. El código de programación para establecer esta comunicación se detalla en el 7 Programación.

### 5.3.1. Comunicación SPI

La forma de comunicación se establece mediante el uso de 3 pines entre el maestro y todos los dispositivos o esclavos, estos son:

- Serial Clock (SCLK)
- Master Out Slave In (MOSI), para el envío de información del maestro al esclavo.
- Master In Slave Out (MISO), para el envío de información del esclavo al maestro.

Cada esclavo necesita además un pin adicional para la selección de estos. Este es conocido como Slave Select (SS). Por lo tanto, el número total de pines I/O necesarios será 3+n, siendo n el número de esclavos. Gráficamente podemos ver a continuación un esquema de cómo se configuran las conexiones:

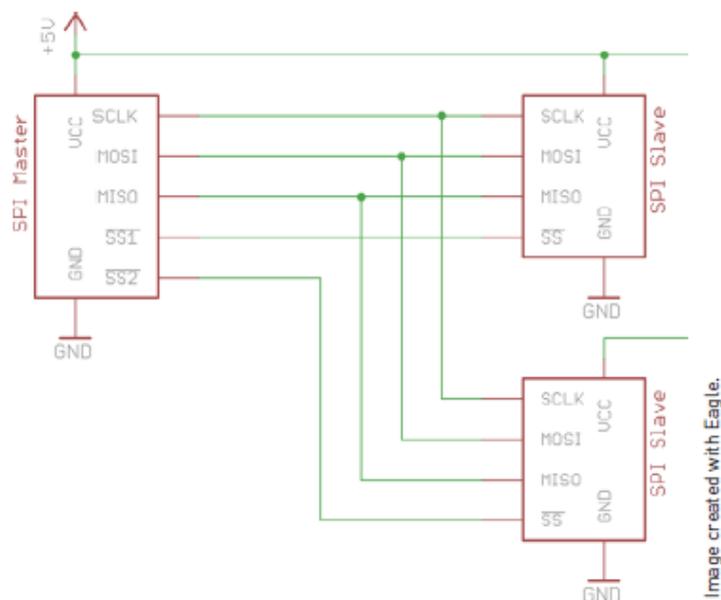


Figura 60. Configuración de conexiones (SPI)

Las funciones de cada línea se describen a continuación:

- Shared/Serial Clock (SCLOCK): en esta línea, el maestro envía pulsos de reloj. Cada flanco de subida de cada pulso determina el momento en el que se transmite información entre el esclavo y el maestro, por lo que cada pulso se corresponde con un bit.
- Master Out Slave In (MOSI): esta línea es la de comunicación de maestro a esclavo: el maestro fija el valor binario en su salida MOSI que es leído en la correspondiente entrada MOSI, mediante una configuración de alta impedancia de entrada, en el esclavo.
- Master In Slave Out (MISO): esta línea es la de comunicación de esclavo a maestro: el esclavo fija el valor binario en su salida MISO que es leído en la correspondiente entrada MISO, mediante una configuración de alta impedancia de entrada, en el maestro.
- Slave Select (SS): hay tantas líneas SS como esclavos tenemos. Puesto que para esta aplicación únicamente contamos con uno, que es el convertidor, tendremos sólo 4 líneas que salen del maestro. Esta línea está normalmente en un valor de tensión alto, y se pone a un valor de tensión bajo para indicar al esclavo que se está estableciendo comunicación con éste. Por lo tanto, cuando el SS está en valor alto, el esclavo obvia cualquier tipo de información que llegue del maestro.

Una vez definida cada línea, se establecen las conexiones haciendo uso de los pines que ya están configurados para establecer una comunicación SPI tanto en el ESP8266 como en el MCP3304. En la siguiente figura se muestra cómo se han conectado:

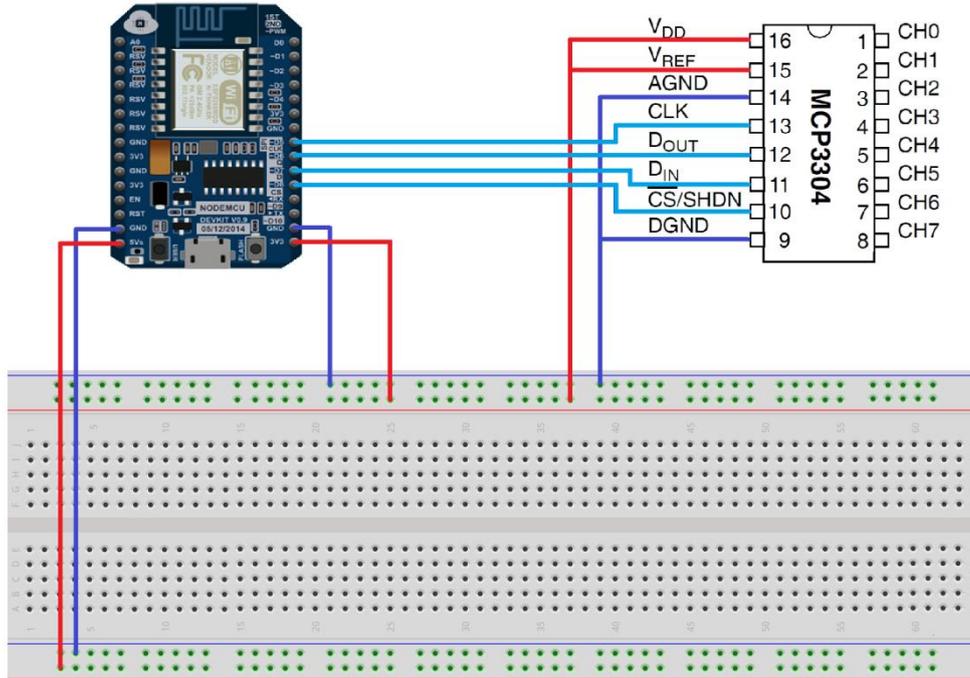


Figura 61. Configuración de conexiones (SPI)

Donde el pin MISO del microcontrolador, se conecta al  $D_{OUT}$  del convertidor, puesto que es la línea que transmite datos desde el convertidor (esclavo) al microcontrolador (maestro). Y el pin MOSI del microcontrolador, se conecta al  $D_{IN}$  del convertidor, puesto que la transmisión de datos se da desde el microcontrolador al convertidor.

A continuación, se realizan pruebas con el código detallado en *7Programación*. En una fase inicial, se realizan pruebas configurando el convertidor para la lectura de señales del tipo single-ended. Asimismo, como primera aproximación, las pruebas se realizarán con el MCP3008.

De esta forma, mediante el uso de un osciloscopio, captaremos las señales de las 4 líneas implicadas en la conexión SPI.

La primera experiencia se realizará tomando como entrada la alimentación, esto es 3,3 V. Así pues, el convertidor ADC muestreará el mayor valor posible. Teniendo en cuenta la resolución del MCP3008, que es de 10 bits, este valor en digital será  $2^{10} - 1$ , esto es 1023. Este es el valor que esperamos obtener.

Este montaje se representa en la siguiente figura:

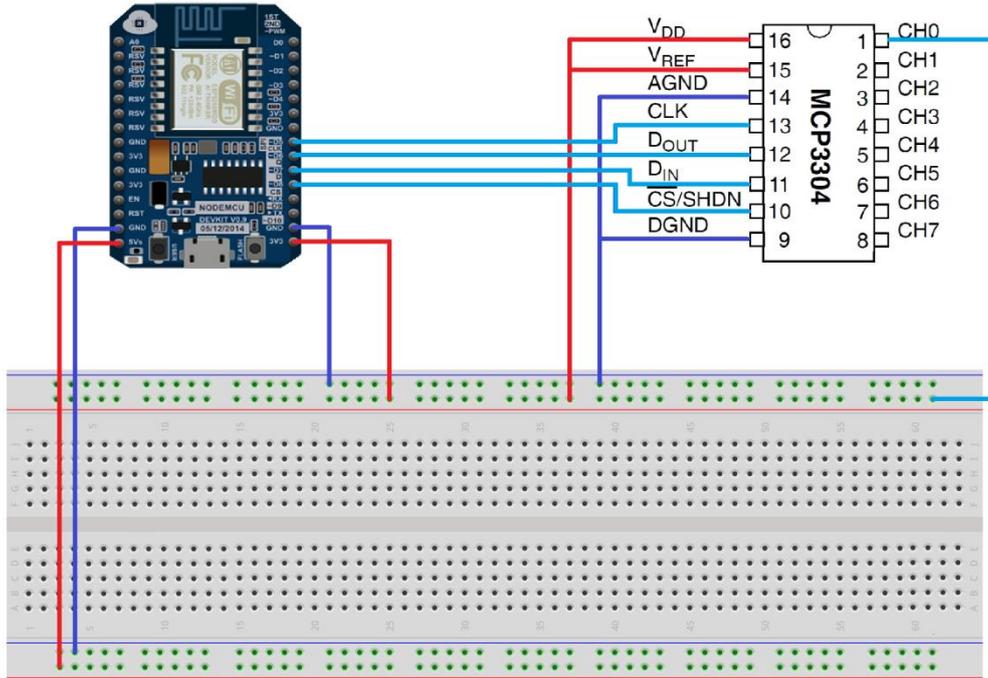


Figura 62. Configuración de conexiones (SPI)

Haciendo uso del osciloscopio, se procede a obtener las señales de las 4 líneas, teniendo en cuenta que estamos muestreando el valor de tensión de la alimentación con el canal 0. Éstas se muestran gráficamente a continuación:

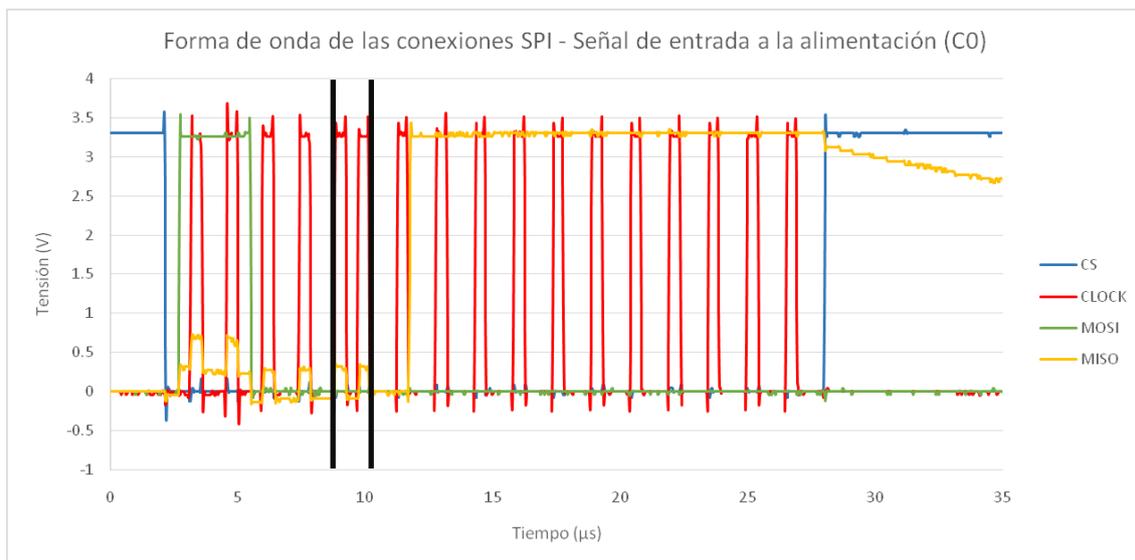


Figura 63. Comunicación SPI

Como se introdujo en el punto anterior Transmisión de datos, la comunicación SPI se da de la siguiente manera:

- En primer lugar, para establecer la comunicación con el esclavo, la línea CS se debe poner a nivel bajo, para que el esclavo entre en modo de escucha y procese la información que le llega del maestro.
- A continuación, el maestro se comunica con el esclavo, de forma que en cada pulso de reloj envía un bit, siguiendo este criterio:
  - El primer pulso de reloj constituye el bit de comunicación, y para establecer la comunicación, la señal MOSI estará en valor alto.
  - En el segundo pulso de reloj, el maestro envía información sobre si la señal que pide es del tipo single-ended o del tipo diferencial. En esta prueba, como se puede apreciar está en valor alto, que se corresponde con single-ended, que es el modo usado para esta prueba como se indicó anteriormente.
  - En los próximos 3 pulsos de reloj el maestro indica sobre qué entrada del convertidor está pidiendo información. Notar que el convertidor tiene 8 entradas analógicas posibles, que se corresponden con 3 bits, pues tenemos que con 3 bits podemos representar  $2^3$  valores. En este ejemplo, vemos que la señal MOSI está a nivel bajo en estos 3 bits, puesto que estamos muestreando con el canal 0.
  - El convertidor empieza a muestrear en el flanco de subida del último de los 3 bits que indican el canal seleccionado, puesto que justo en este momento es conocido el canal que se debe muestrear. El tiempo de muestreo finaliza en el flanco de bajada del siguiente pulso de reloj, con lo que hay un pulso de reloj en el que el maestro no transmite ni recibe información. Este tiempo es de 1,9  $\mu$ s y será el tiempo utilizado en el 4.2.1.1 *Divisor de tensión diferencial*.
- Tras el muestreo de la señal, tenemos un pulso de reloj sin información, que marca el momento en el que se cambia el sentido de transmitir la información.
- Tras este bit nulo, le siguen una serie de bits que se transmiten en cada pulso que se corresponde al valor de la señal analógica (convertida a digital) de la dirección (pin o patilla de entrada analógica) especificada por el maestro. Este número de bits dependerá de la resolución deseada, siendo en este caso una resolución de 10 bits. Como podemos ver gráficamente, todos los bits están en valor alto, que es lo que cabía esperar teniendo en cuenta que estamos muestreando la señal de la alimentación, esto es 3,3 V, y el valor mostrado por el microcontrolador se corresponde con 1023, como se había indicado.

Realizando más pruebas, podemos ver qué se transmite si en lugar de conectar el canal 0 a la alimentación, lo conectamos a tierra. De esta forma, el montaje quedaría:

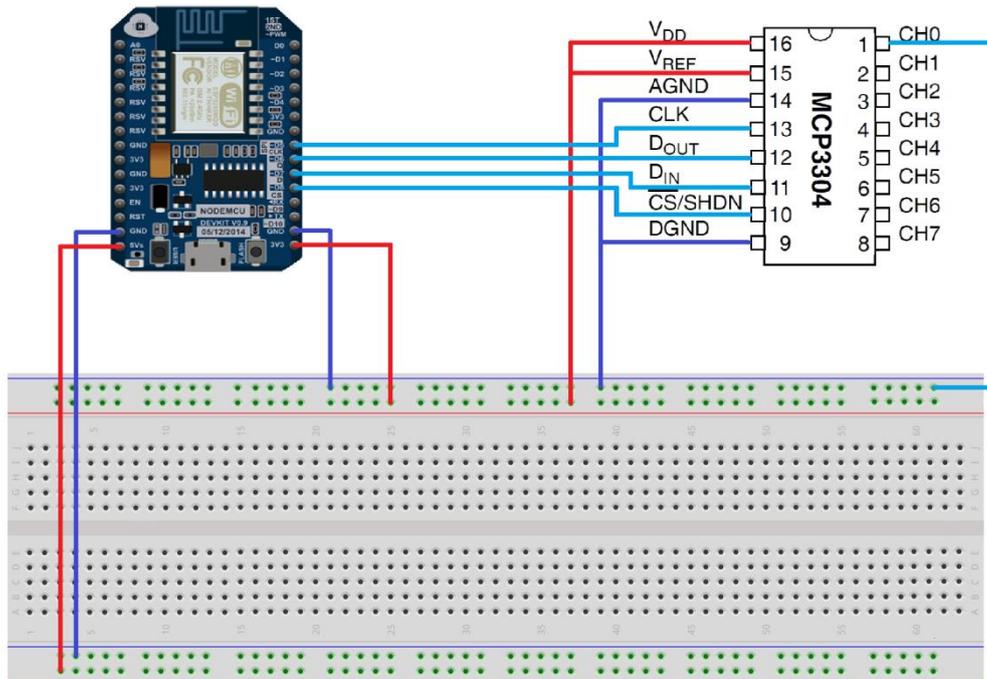


Figura 64. Configuración de conexiones (SPI)

Gráficamente lo que obtenemos es lo siguiente:

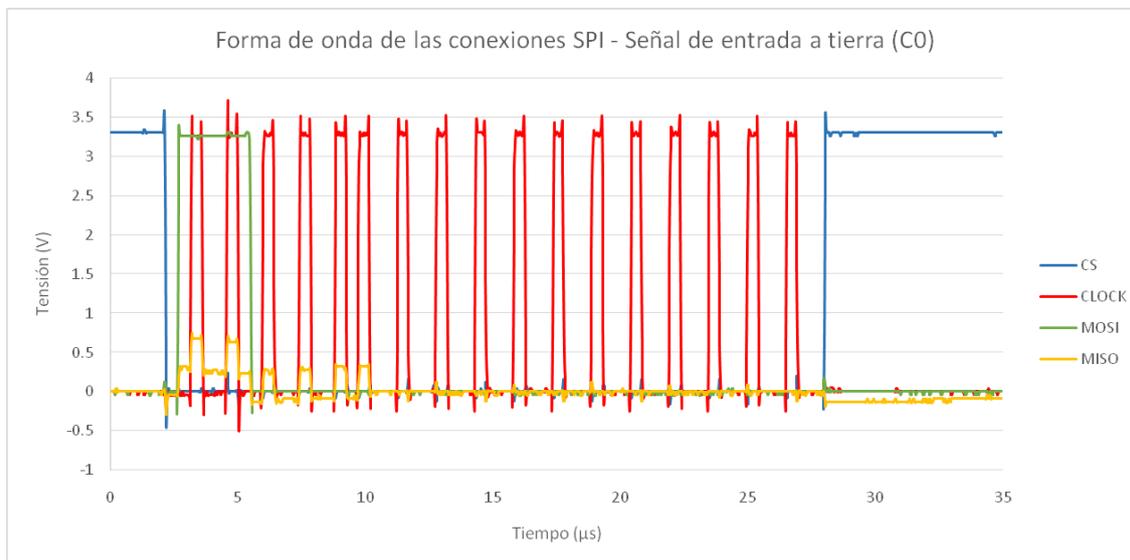


Figura 65. Comunicación SPI

En este caso, como se puede apreciar, respecto al caso anterior tenemos que para la línea MISO, en los bits de envío del valor muestreado, están todos a nivel bajo. Luego el valor que se está transmitiendo es 0.

A continuación, pasamos a muestrear valores de tensión que están entre 0 V y 3,3 V. Esto lo conseguimos mediante el uso de un divisor de tensión entre la alimentación y tierra. El montaje para este caso quedaría:

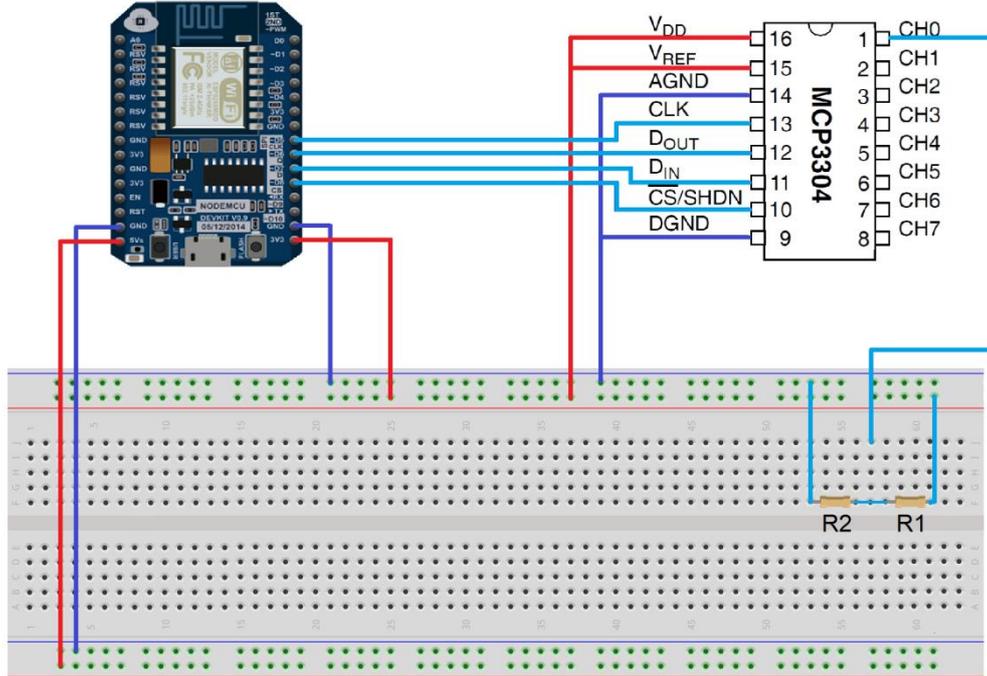


Figura 66. Configuración de conexiones (SPI)

Para el divisor de tensión se utilizarán resistencias de 3,9 kΩ, 4,3 kΩ y de 6,2 kΩ, haciendo combinaciones para obtener diferentes parejas para la realización de pruebas. En primer lugar se seleccionarán como parejas 3,9 kΩ y 4,3 kΩ, de forma que la tensión muestreada por el convertidor será:

$$V_{out,analog} = V_{in} \cdot \frac{R_2}{R_1 + R_2} = 3,3 \cdot \frac{4,3}{3,9 + 4,3} = 1,73 \text{ V} \quad (\text{LIV.})$$

Que en digital se corresponde con un valor de:

$$V_{out,dig} = V_{out,analog} \cdot \frac{1023}{3,3} = 536 \quad (\text{LV.})$$

Que en binario es 100011000.

En el osciloscopio, vemos que las señales capturadas son las siguientes:

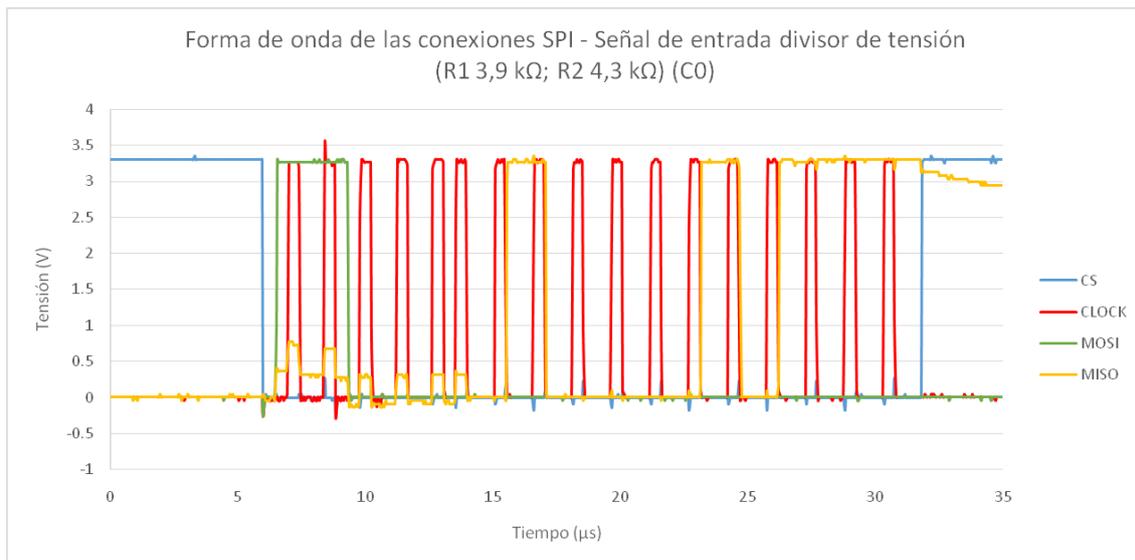


Figura 67. Comunicación SPI

Donde se está enviando en binario 1000010111, que se corresponde en decimal con 535, que por comparación con 536, se puede dar como válido (recordemos que la resolución en este equipo es de 10 bits, que corresponden a 1/1024 o 0,1%, menor que la precisión de las resistencias utilizadas en el divisor de tensión).

Si seleccionamos como parejas 3,9 kΩ y 6,2 kΩ, la tensión muestreada por el convertidor será:

$$V_{out,analog} = V_{in} \cdot \frac{R_2}{R_1 + R_2} = 3,3 \cdot \frac{6,2}{3,9 + 6,2} = 2,03 \text{ V}$$

(LVI.)

Que en digital se corresponde con un valor de:

$$V_{out,dig} = V_{out,analog} \cdot \frac{1023}{3,3} = 628$$

(LVII.)

Que en binario es 1001110100.

En el osciloscopio, vemos que las señales capturadas son las siguientes:

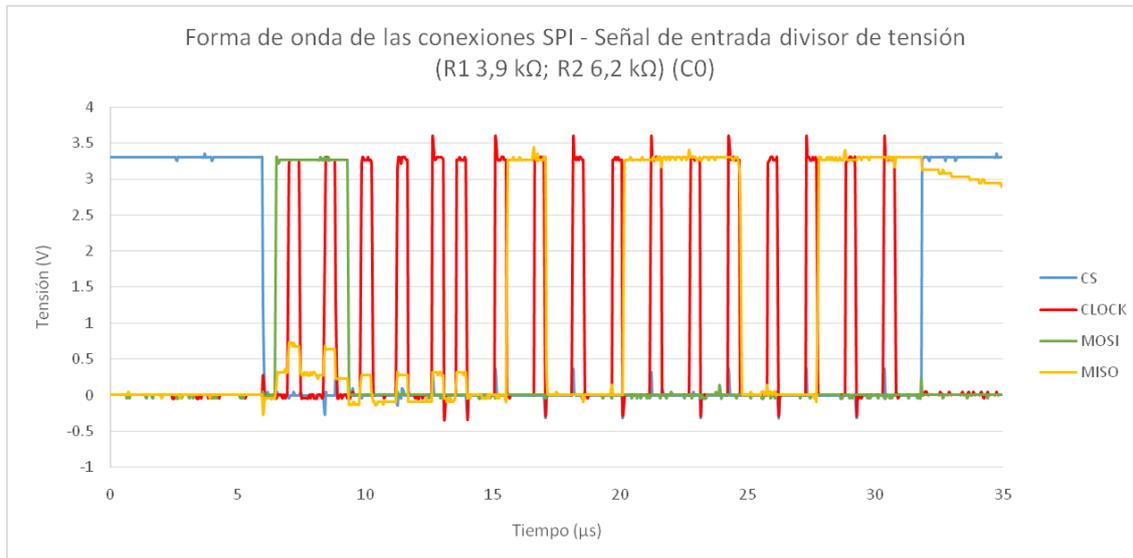


Figura 68. Comunicación SPI

Los datos que envía el convertidor en binario es 1001110011, que en decimal es 627, que por comparación con 628, se da como válido.

Por último, si seleccionamos como parejas 4,3 kΩ y 6,2 kΩ, la tensión muestreada por el convertidor será:

$$V_{out,analog} = V_{in} \cdot \frac{R_2}{R_1 + R_2} = 3,3 \cdot \frac{6,2}{4,3 + 6,2} = 1,95 \text{ V}$$

(LVIII.)

Que en digital se corresponde con un valor de:

$$V_{out,dig} = V_{out,analog} \cdot \frac{1023}{3,3} = 604$$

(LIX.)

Que en binario es 001001011100.

En el osciloscopio, vemos que las señales capturadas son las siguientes:

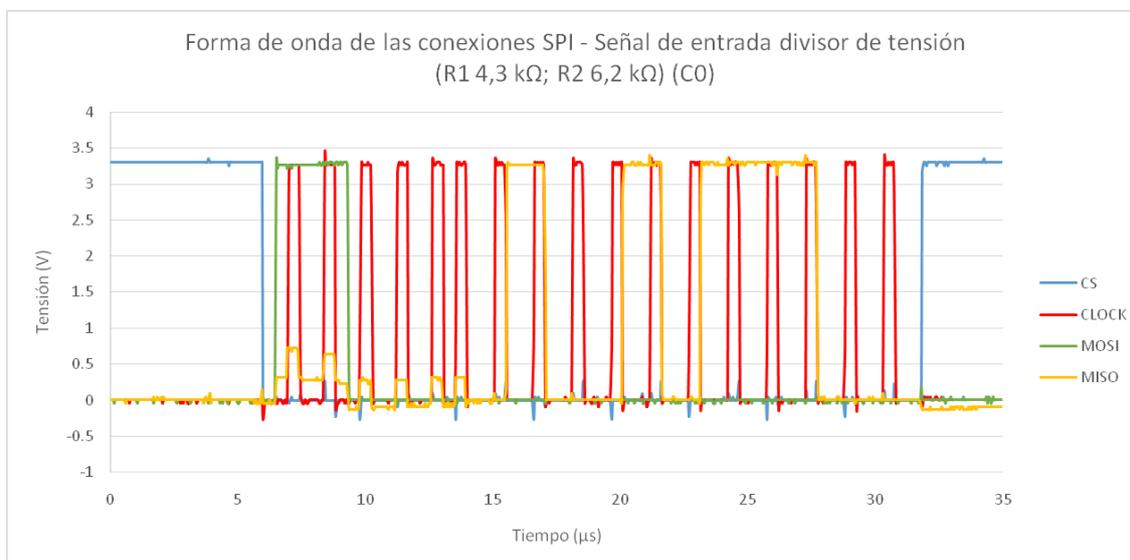


Figura 69. Comunicación SPI

Los datos que envía el convertidor en binario es 1001011100, que en decimal es 604, igual que el valor esperado.

Por otro lado, pasando a otro tipo de montaje, donde cambiamos el canal de donde se muestra, pasando a ser el canal 1, el montaje sería:

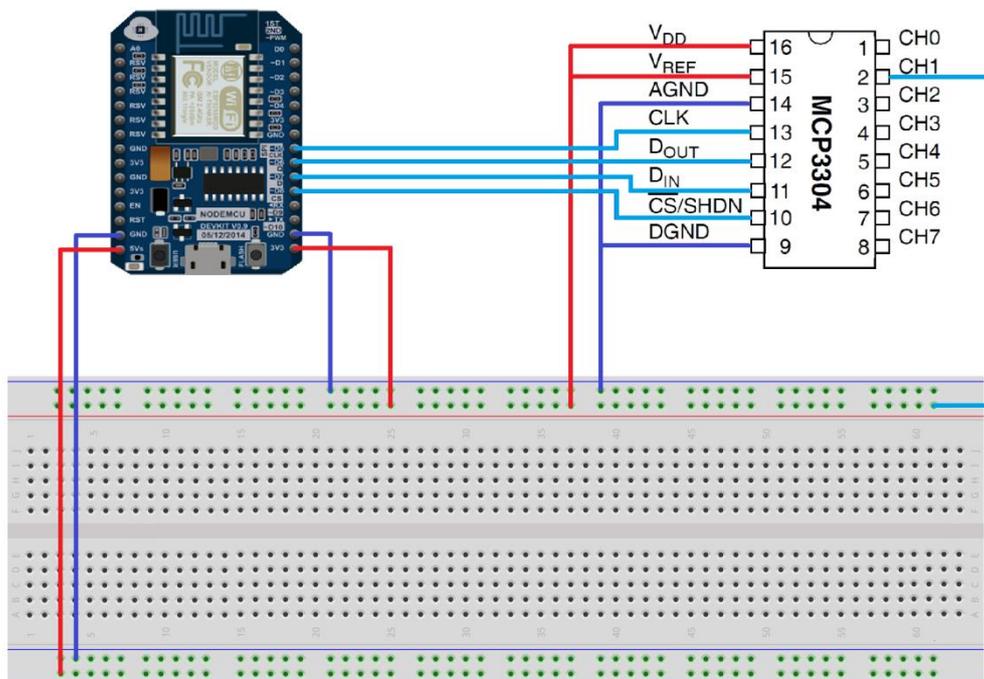


Figura 70. Configuración de conexiones (SPI)

Y del osciloscopio, obtenemos las siguientes señales de las líneas:

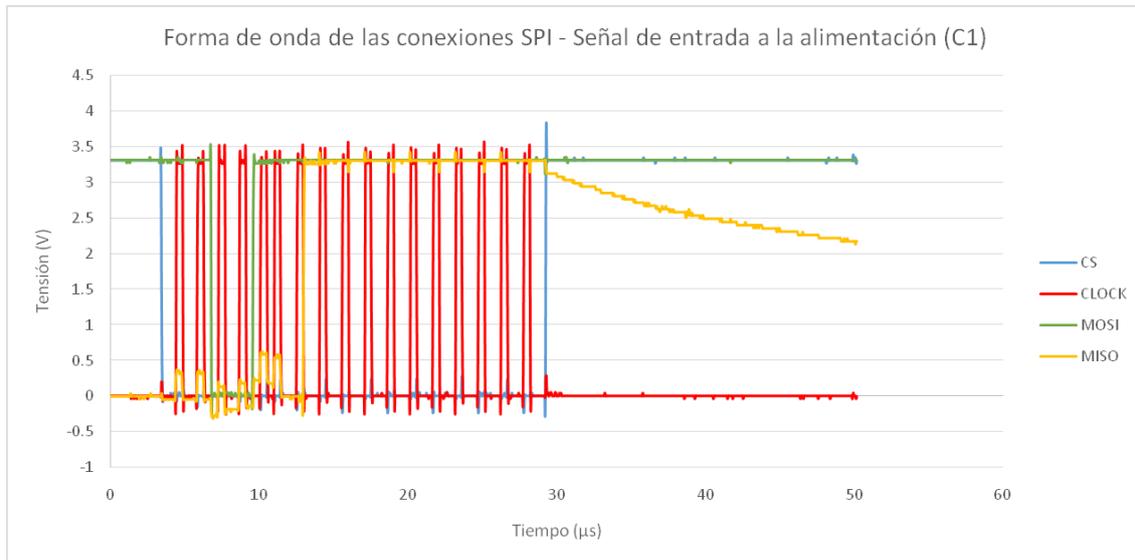


Figura 71. Comunicación SPI

Donde vemos que los 3 bits de la línea MOSI que indican la dirección del canal, en este caso transmite 001, que se corresponde con el decimal 1.

Si modificamos el montaje, para leer del canal 6, tenemos:

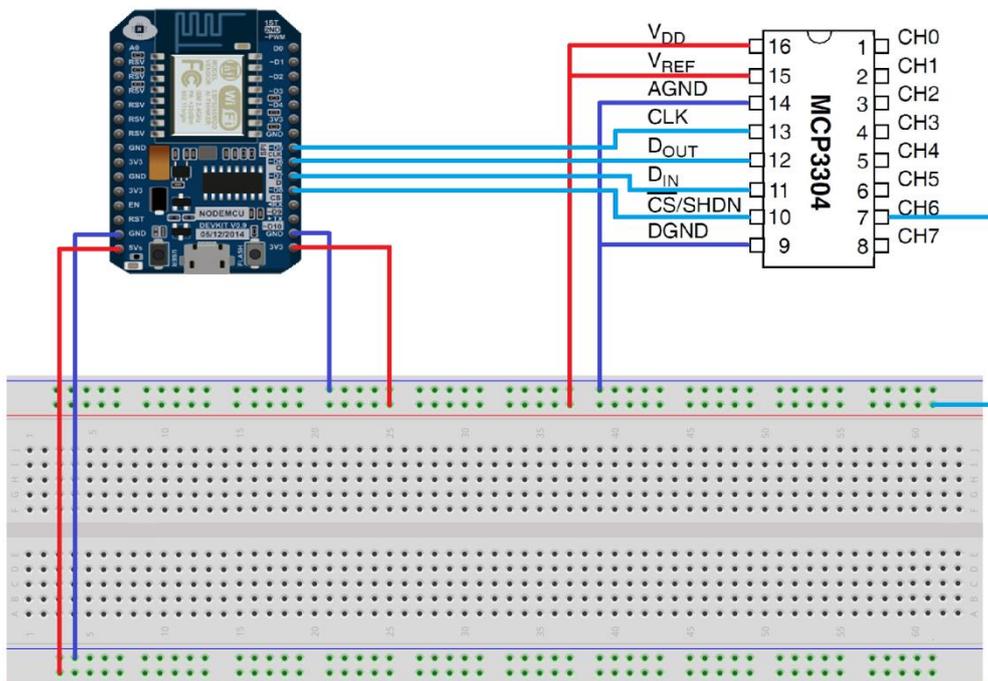


Figura 72. Configuración de conexiones (SPI)

Y con el osciloscopio obtenemos:

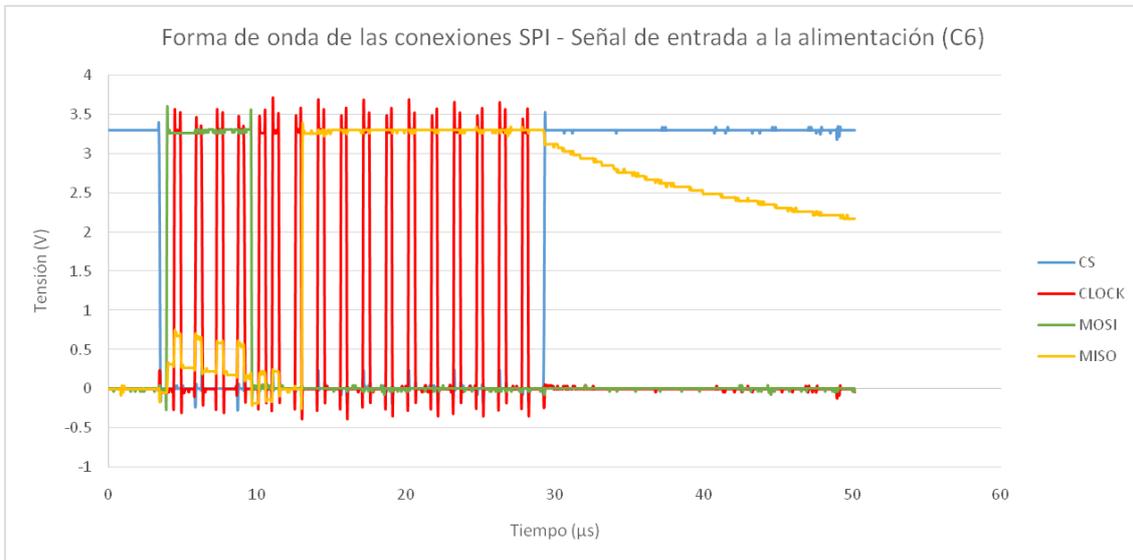


Figura 73. Comunicación SPI

En esta ocasión, indica la dirección 110, que en decimal es 6.

Finalmente, probando el canal 7, el montaje queda:

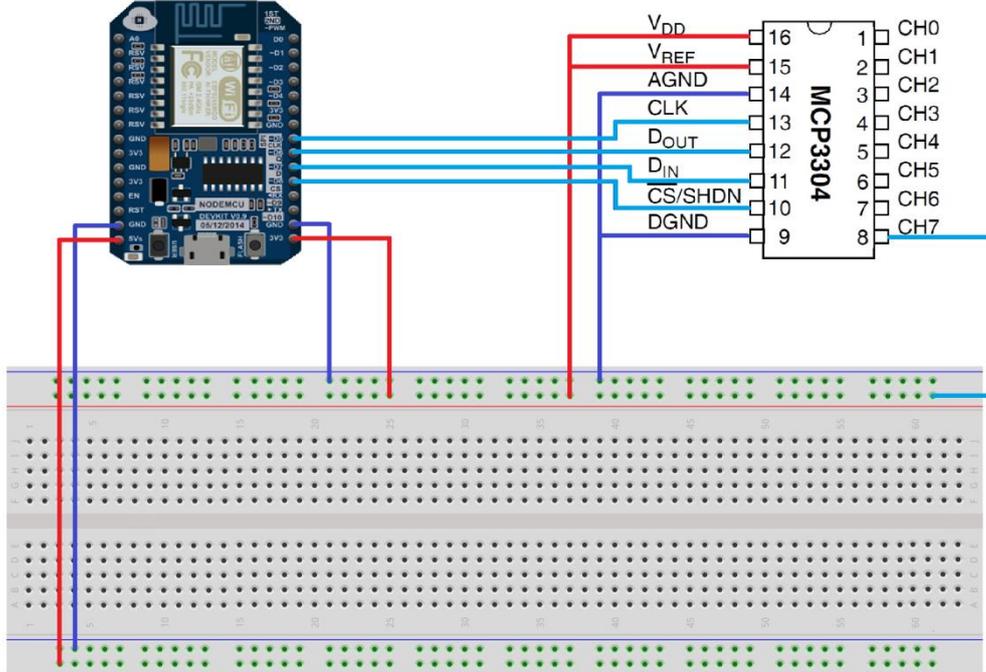


Figura 74. Configuración de conexiones (SPI)

Y del osciloscopio obtenemos:

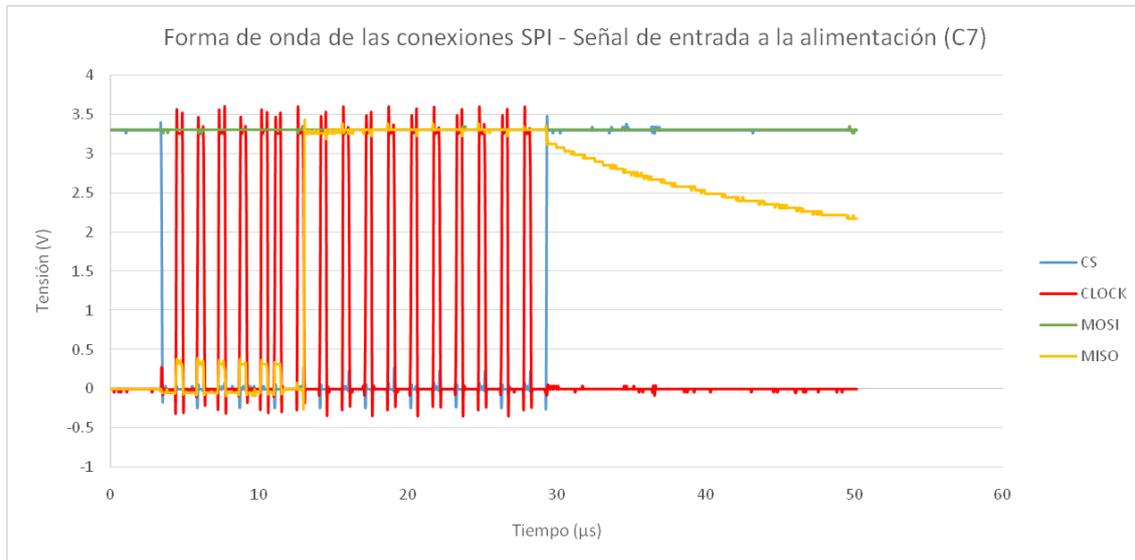


Figura 75. Comunicación SPI

Que indica la dirección 111, que en decimal es 7.

Para finalizar, se realizarán dos muestreos consecutivos, puesto que, en la práctica, pretendemos medir tensión y corriente. Para este ejemplo, muestrearemos por el canal 0 y 1, teniendo por el canal 0 un divisor de tensión formado por una pareja de resistencias de 3,9 k $\Omega$  y 4,3 k $\Omega$ , y en el canal 1, muestreamos la tensión de alimentación. De esta forma, el montaje quedaría:

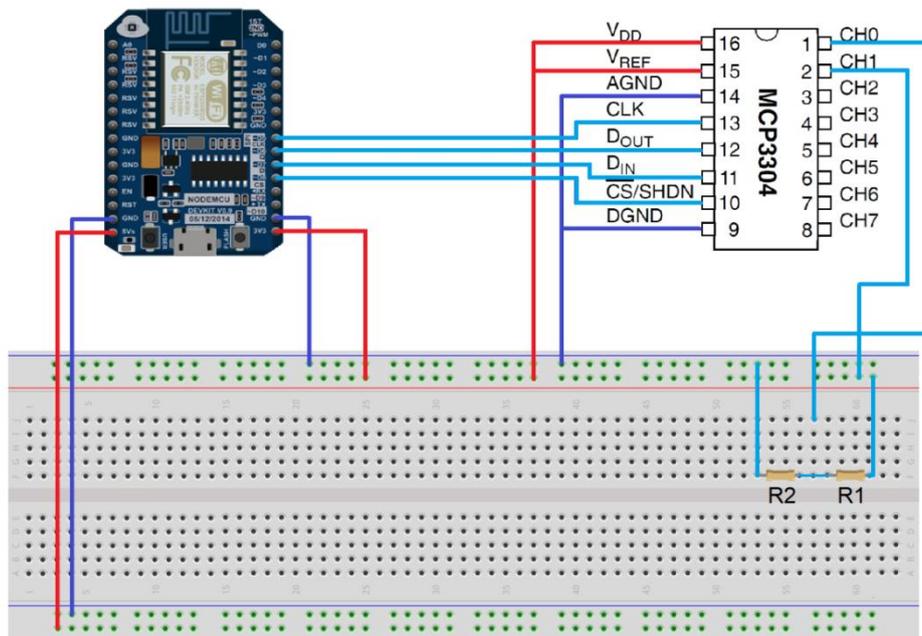


Figura 76. Configuración de conexiones (SPI)

Capturando las señales con el osciloscopio, tenemos:



Figura 77. Comunicación SPI

De esta prueba, podemos extraer que para la transmisión de datos consecutivos, tenemos que se tarda aproximadamente  $15 \mu\text{s}$ , lo que resulta un tiempo muy pequeño en comparación con el ciclo de onda de la tensión ( $15 \mu\text{s}$  equivalen a un ángulo eléctrico de  $0,27^\circ$  para señales de 50 Hz), por lo que en la práctica se puede considerar que ambas muestras consecutivas se dan en el mismo instante de tiempo. Esto nos permitirá calcular la potencia activa como producto del valor instantáneo de tensión y corriente.

## 6. Determinación del precio de la energía

Como ya se ha introducido en el primer capítulo, el fin último de este proyecto es la determinación del precio de la energía consumida por el usuario. Hasta este punto, se ha tratado todo el desarrollo de la instrumentación para ser capaces de obtener el consumo eléctrico.

Así pues, somos capaces de desarrollar un montaje que proporciona el valor de la potencia consumida (W) a lo largo de un minuto, una hora, o incluso días. Donde se considera que la potencia consumida a lo largo de ese minuto es constante, y para el caso del precio de la energía a lo largo de una hora y de un día, se determinará una potencia media a lo largo de esa hora, mediante la lectura de las potencias minuto a minuto. De esta forma, el precio de la energía consumida, se puede calcular como:

$$\begin{aligned}
 & \text{Coste de la electricidad histórico minutos(€)} \\
 & = \text{Consumo eléctrico (kW)} \cdot \text{Precio energía eléctrica} \left( \frac{\text{€}}{\text{kWh}} \right) \cdot 1 \text{ min} \cdot \frac{1 \text{ h}}{60 \text{ min}}
 \end{aligned}
 \tag{LX.}$$

$$\begin{aligned}
 & \text{Coste de la electricidad histórico horas(€)} \\
 & = \text{Consumo eléctrico (kW)} \cdot \text{Precio energía eléctrica} \left( \frac{\text{€}}{\text{kWh}} \right) \cdot 60 \text{ min} \cdot \frac{1 \text{ h}}{60 \text{ min}}
 \end{aligned}
 \tag{LXI.}$$

$$\begin{aligned}
 & \text{Coste de la electricidad histórico días(€)} \\
 & = \text{Consumo eléctrico (kW)} \cdot \text{Precio energía eléctrica} \left( \frac{\text{€}}{\text{kWh}} \right) \cdot 1440 \text{ min} \cdot \frac{1 \text{ h}}{60 \text{ min}}
 \end{aligned}
 \tag{LXII.}$$

De esta forma, sólo queda conocer el precio de la energía eléctrica para poder determinar el precio de la energía consumida por el usuario. Para esto, podemos obtener la información del precio de la energía a través del Sistema de Información del Operador del Sistema<sup>(21)</sup>, donde según el Real Decreto 216/2014<sup>(22)</sup>, establece que aquellos consumidores de baja tensión con una potencia contratada de hasta 10 kW son los que se pueden acoger al Precio Voluntario para el Pequeño Consumidor (PVPC). Puesto que el presente proyecto va destinado al consumo doméstico, se centrará en el estudio del precio de la energía para el caso del PVPC, donde encontramos tres tipos de tarifas:

- Tarifa con precio único. Tenemos la tarifa por defecto, o peaje 2.0 A, la cual está definida por un precio único de la energía.
- Tarifa de dos periodos. En este caso tenemos la tarifa eficiencia 2 periodos, o peaje 2.0 DHA, la cual viene definida por dos precios de la energía en distintas franjas horarias, según estemos en horario de verano o invierno, y la tarifa de vehículo eléctrico, o peaje 2.0 DHS, en la cual se tiene un precio de la energía ligeramente inferior durante las horas nocturnas, pensado para aquellos propietarios de vehículos eléctricos donde

en las horas de carga de éste (durante la noche), la energía tiene un precio ligeramente inferior para ahorrar costes.

Los periodos tarifarios, para la 2.0 DHA y la 2.0 DHS, se definen en el “*Real Decreto 647/2011*”<sup>(23)</sup> y en la “*Orden ITC/2794/2007*”<sup>(24)</sup>, por la que se revisan las tarifas eléctricas. Estos periodos se distribuyen:

Invierno		Verano	
Punta	Valle	Punta	Valle
12-22	0-12 22-24	13-23	0-13 23-24

Figura 78. Periodos tarifa 2.0 DHA

Invierno y Verano		
P1	P2	P3
13-23	0-1 7-13 23-24	1-7

Figura 79. Periodos tarifa 2.0 DHS

Donde, como podemos apreciar, en la tarifa 2.0 DHS existe un tercer periodo “supervalle” donde el precio es ligeramente inferior como veremos a lo largo del desarrollo de este capítulo.

Como se establece en el *Real Decreto 216/2014*<sup>(22)</sup>, el precio de la energía, se calcula como:

$$FEU = \sum_{\text{periodo facturación}} \left[ (E_p \cdot TEU_p) + \sum_{h \in p} (E_{p_h} \cdot TCU_h) \right]$$

Donde:

$E_p$ : Es la energía consumida en el período tarifario p expresado en kWh

$TEU_p$ : Es el precio del término de energía del PVPC del período tarifario p, expresado en €/kWh

$E_{p_h}$ : Es la energía consumida en la hora h del período tarifario p, expresado en kWh

$TCU_h$ : Es el precio del término de coste horario de energía del PVPC, en cada hora h, expresado en €/kWh. Donde  $TCU_h$  se calcula como:

$$TCU_h = (1 + PERD_h) \cdot CPh$$

Donde:

$PERD_h$ : Es el coeficiente de pérdidas del peaje de acceso de aplicación al suministro en la hora  $h$

$CP_h$ : Es el coste de producción de la energía suministrada en cada hora expresado en €/kWh

En la práctica, Red Eléctrica de España (REE), a través del Sistema de Información del Operador del Sistema(21), calcula el precio horario a aplicar, de forma que el coste de la energía se determina con el precio publicado por REE como:

$$FEU = \sum EP_h \cdot PE_h$$

Donde:

$EP_h$ : Es la energía consumida en cada hora  $h$

$PE_h$ : Es el precio total horario de la energía, que incluye:

- El término de la tarifa de acceso, publicado en el BOE
- El precio horario de la energía
- El coste de la energía asociado a las pérdidas del sistema
- Los servicios de ajuste y otros costes del sistema.

Si descargamos estos precios de la energía para un año natural, esto es, desde el 1 de enero del 2017, y hasta el 31 de diciembre del 2017, podemos estimar un precio de la energía para cada tarifa. Así pues, gráficamente, el precio de la energía a lo largo de un año para cada tipo de tarificación es:

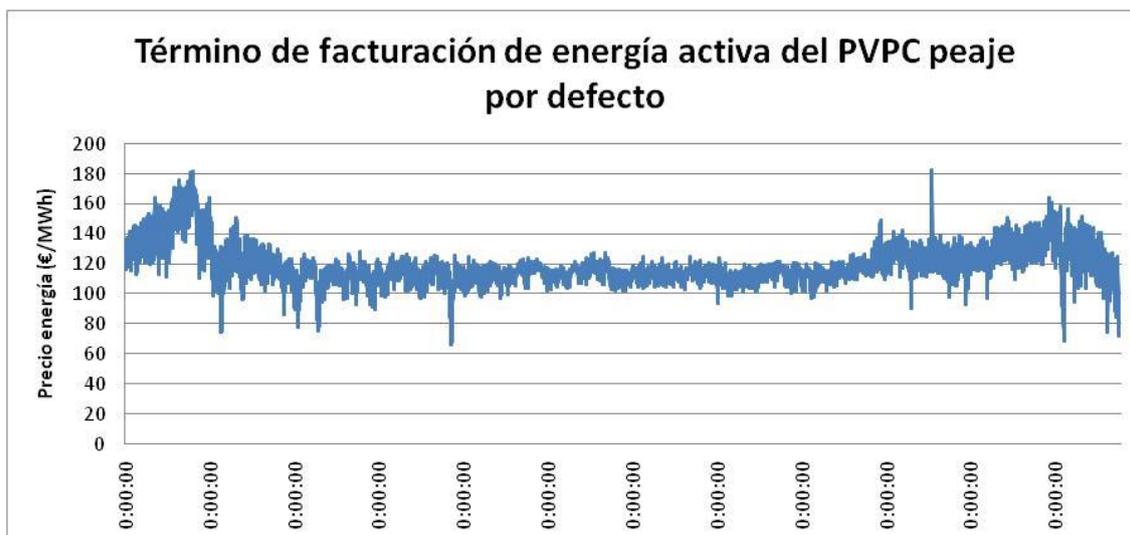


Figura 80. Término de facturación de energía activa del PVPC peaje por defecto

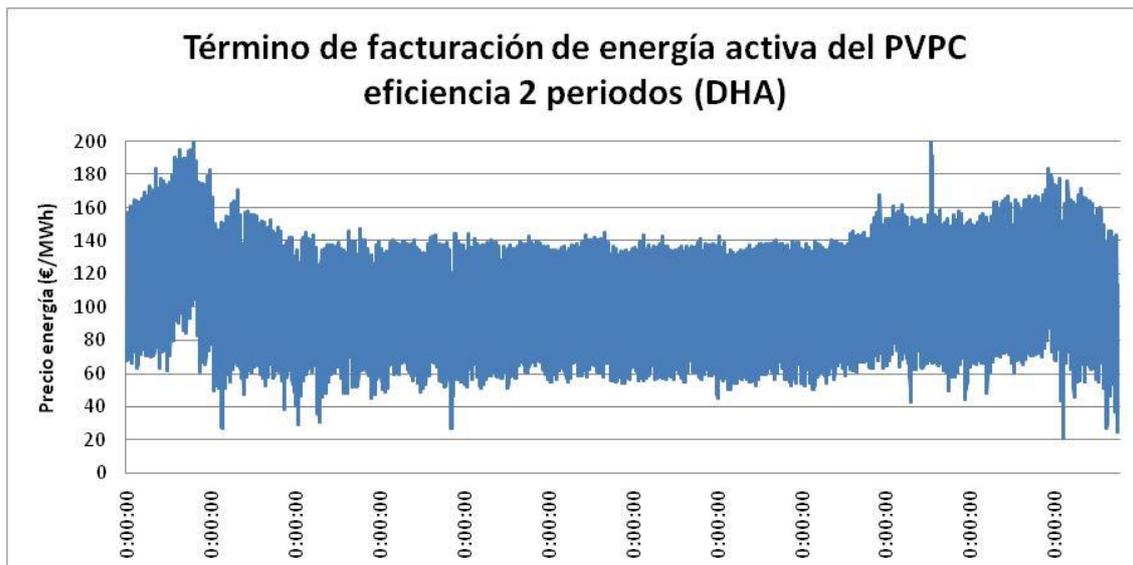


Figura 81. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA)

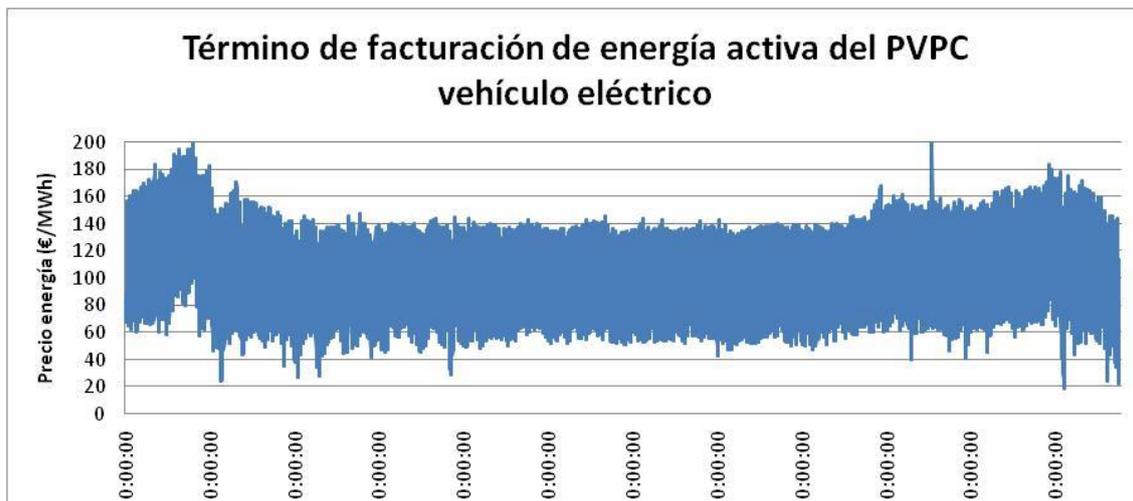


Figura 82. Término de facturación de energía activa del PVPC vehículo eléctrico

Como primera aproximación, podemos ver que el peaje por defecto, no presenta una gran variabilidad si los comparamos con las tarifas de dos periodos, pues éste consta de un único período y un único precio del término de la energía, aunque el coste total de la energía presenta una pequeña variabilidad debido a los costes antes mencionados asociados al precio publicado por REE. Asimismo, de las gráficas, podemos ver que la tendencia de las series temporales es constante, es decir, en el largo plazo, el valor medio del precio de la electricidad es constante. De un análisis más profundo, podemos ver que existe una triple estacionalidad:

- Estacionalidad anual: de las gráficas, podemos ver que existe una estacionalidad anual que coincide con los meses de invierno, con un precio mayor, y con los meses de verano, donde tenemos unos precios menores.

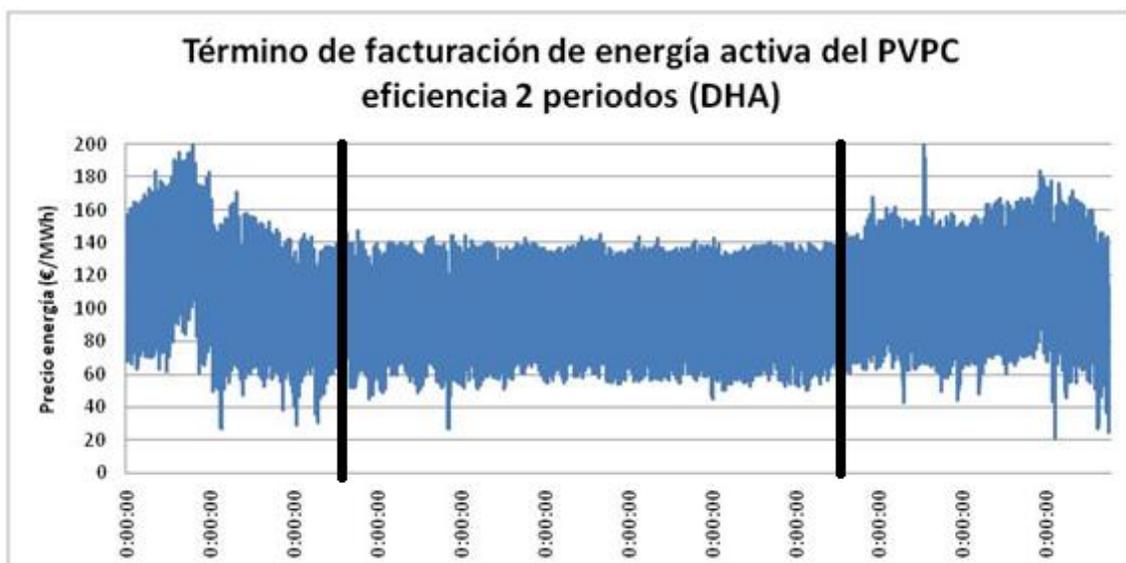


Figura 83. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA) estacionalidad anual

- Estacionalidad semanal: para cada semana, vemos un comportamiento estacional, de forma que para cada semana tenemos un mismo comportamiento, de forma que en los fines de semana el coste es menor.

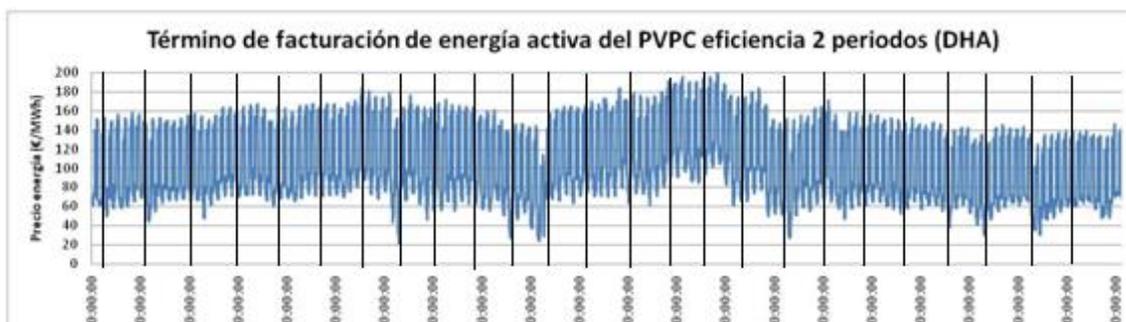


Figura 84. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA): invierno estacionalidad semanal

- Estacionalidad diaria (para las tarifas 2P): a lo largo de cada día, vemos que también existe un comportamiento estacional, donde por las noches el precio de la energía es menor, mientras que durante el día es mayor.

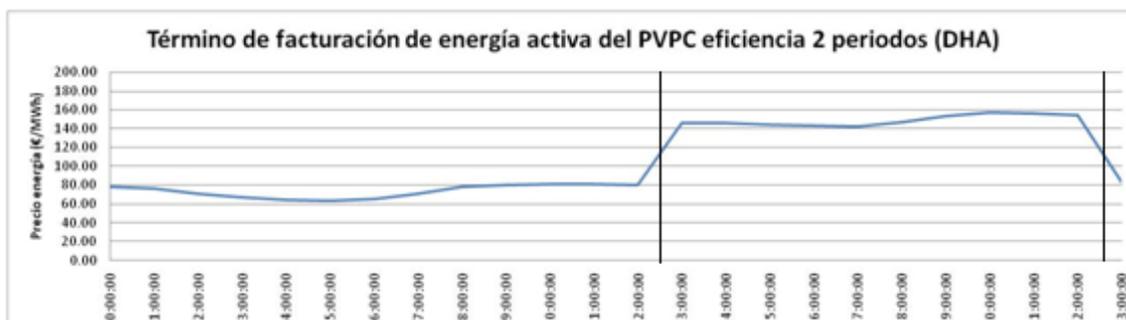


Figura 85. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA): invierno estacionalidad diaria

Así pues, podemos distinguir los meses de verano e invierno, pudiendo calcular un precio total de la energía para cada estación, donde tenemos que la evolución temporal para cada estación y cada tarifa es gráficamente:

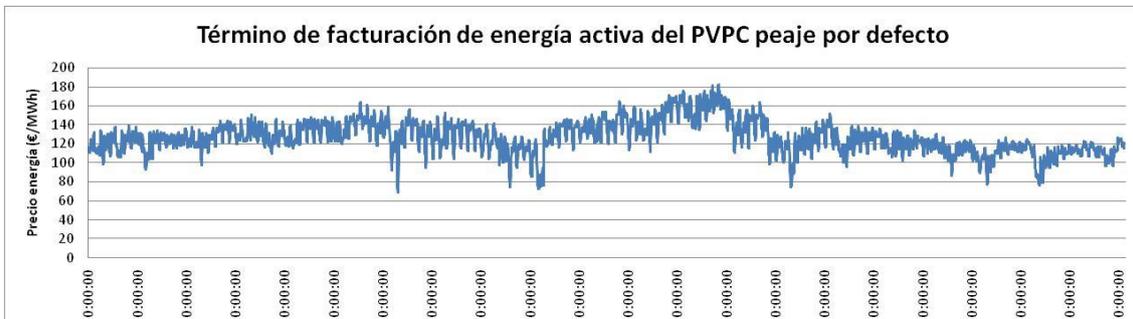


Figura 86. Término de facturación de energía activa del PVPC peaje por defecto: invierno

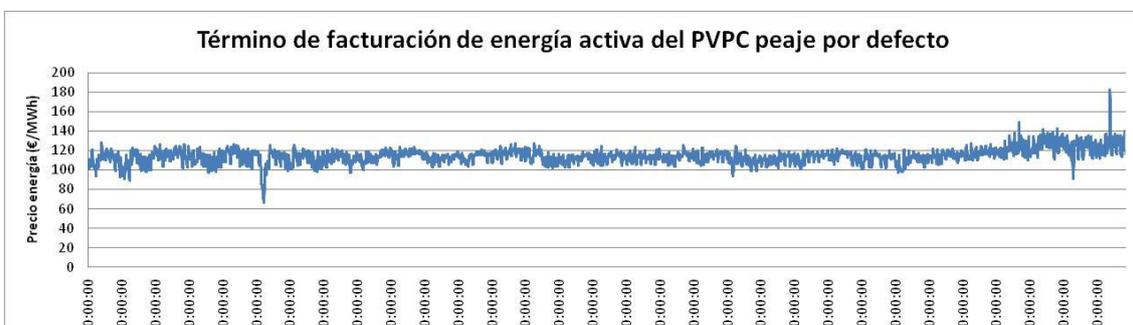


Figura 87. Término de facturación de energía activa del PVPC peaje por defecto: verano

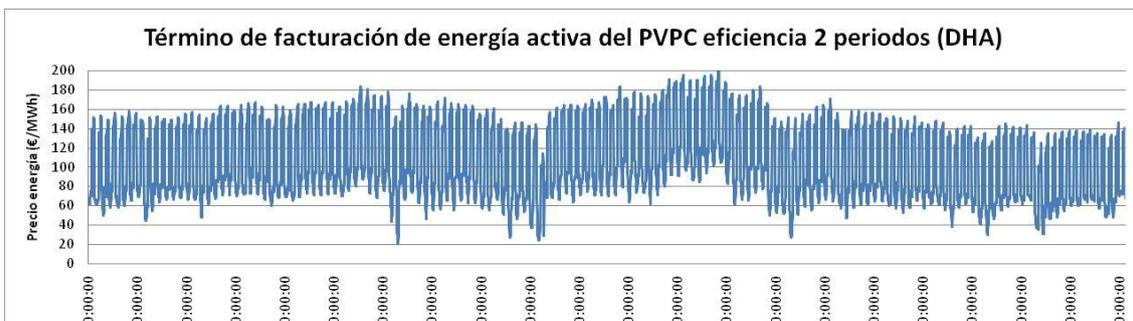


Figura 88. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA): invierno

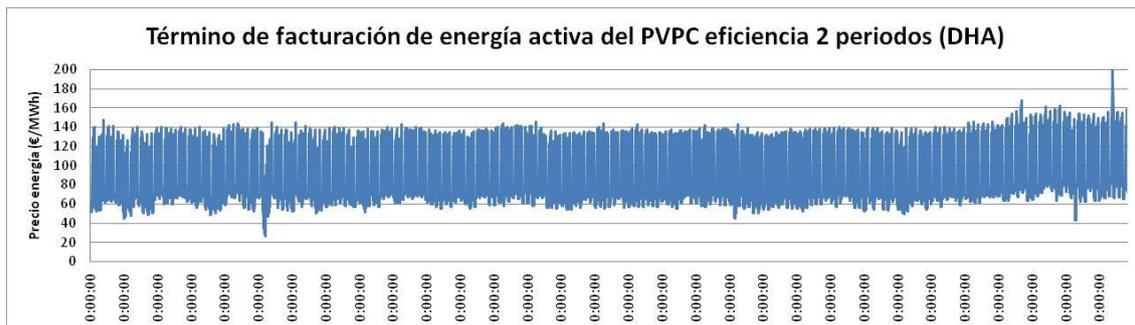


Figura 89. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA): verano

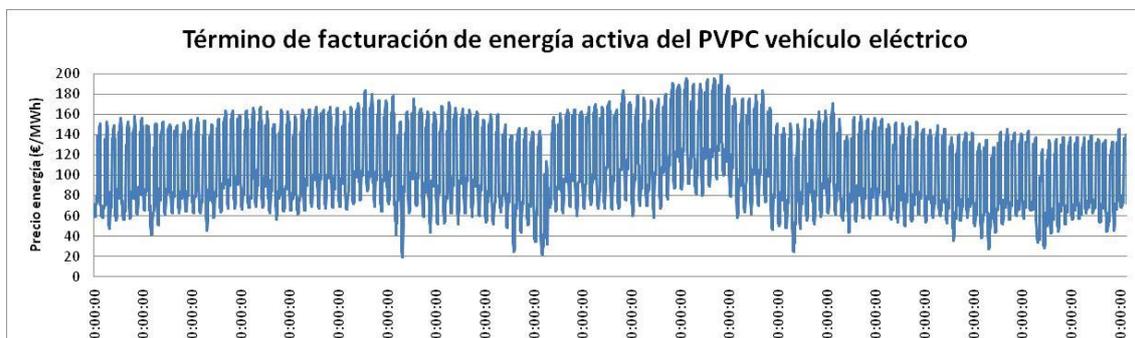


Figura 90. Término de facturación de energía activa del PVPC vehículo eléctrico: invierno

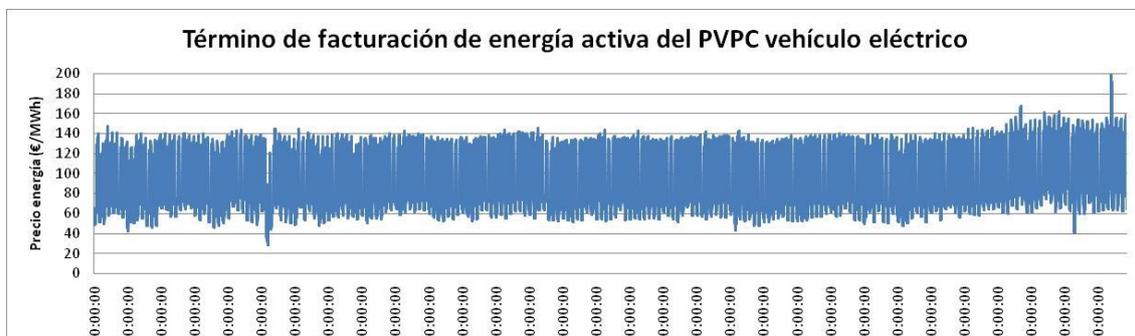


Figura 91. Término de facturación de energía activa del PVPC vehículo eléctrico: verano

Donde se puede ver que la variabilidad del precio de la energía analizando el precio por estaciones es menor.

Esta dispersión en los datos, se puede calcular mediante la desviación típica, que se calcula como:

$$\sigma_i = \sqrt{\frac{\sum_{j=0}^N (\text{Precio energía hora } i_{\text{dia } j} - \overline{\text{Precio energía hora } i})^2}{N}}$$

Donde:

$\sigma_i$ : Es la desviación típica de la hora i

*Precio energía hora  $i_{\text{dia } j}$* : Es el precio de la energía en la hora i, en el día j

$\overline{\text{Precio energía hora } i}$ : Es el precio media de la energía en la hora i

N: Número total de días

Así pues, calculamos la desviación típica para las 24 h, para cada estación y tarifa, teniendo los siguientes resultados (

Tabla 1):

Tabla 1. Desviación típica del precio de la energía

	2.0 A invierno	2.0 A verano	2.0 DHA invierno	2.0 DHA verano	2.0 DHS invierno	2.0 DHS verano
0:00:00	15,49	6,66	14,79	6,44	15,57	6,32
1:00:00	15,46	6,75	14,77	6,52	15,64	6,37
2:00:00	14,86	6,55	14,18	6,33	13,75	6,18
3:00:00	14,36	6,42	13,72	6,21	13,31	6,07
4:00:00	14,46	6,36	13,80	6,14	13,38	6,00
5:00:00	14,35	6,49	13,70	6,30	13,28	6,18
6:00:00	14,10	7,87	13,47	7,66	13,07	7,98
7:00:00	14,83	7,44	14,23	7,35	13,84	7,49
8:00:00	17,22	6,81	16,63	6,77	17,33	6,86
9:00:00	17,29	6,80	16,84	6,73	17,39	6,84
10:00:00	16,39	6,67	16,02	6,60	16,49	6,71
11:00:00	16,00	6,30	15,61	6,27	16,08	6,34
12:00:00	15,84	6,63	15,44	6,66	15,91	8,22
13:00:00	15,68	6,76	15,80	6,79	16,43	6,79
14:00:00	15,69	6,92	15,80	6,95	15,80	6,95
15:00:00	15,79	7,09	15,92	7,11	15,92	7,11
16:00:00	16,06	7,13	16,19	7,15	16,19	7,15
17:00:00	16,10	8,18	16,22	8,19	16,22	8,19
18:00:00	16,93	9,28	17,06	9,33	17,06	9,33
19:00:00	17,09	8,78	17,21	8,84	17,21	8,84
20:00:00	16,07	7,40	16,20	7,46	16,20	7,46
21:00:00	15,64	5,31	15,77	5,35	15,77	5,35
22:00:00	15,27	5,69	15,41	5,49	15,41	7,30
23:00:00	15,75	6,64	15,03	6,42	17,20	6,68

De esta forma, tenemos la desviación típica, medida del grado de dispersión de los datos con respecto al valor promedio, o dicho de otra forma, la variación esperada con respecto a la media aritmética.

Como se puede concluir de los resultados, en los meses de verano, la dispersión es menor que en los meses de invierno. Para interpretar mejor estos resultados, podemos representar gráficamente el precio medio de la energía calculada, para cada hora, estación y tarifa, con el cual se ha calculado la desviación típica. Así pues, gráficamente tenemos, para la tarifa por defecto:

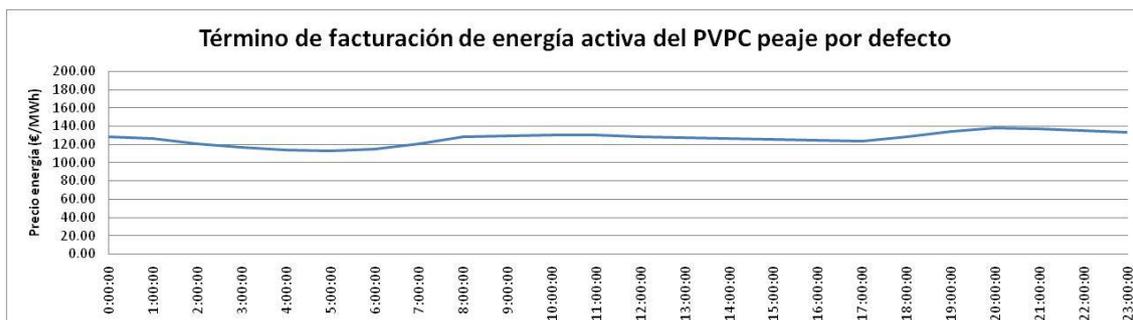


Figura 92. Término de facturación de energía activa del PVPC peaje por defecto: invierno

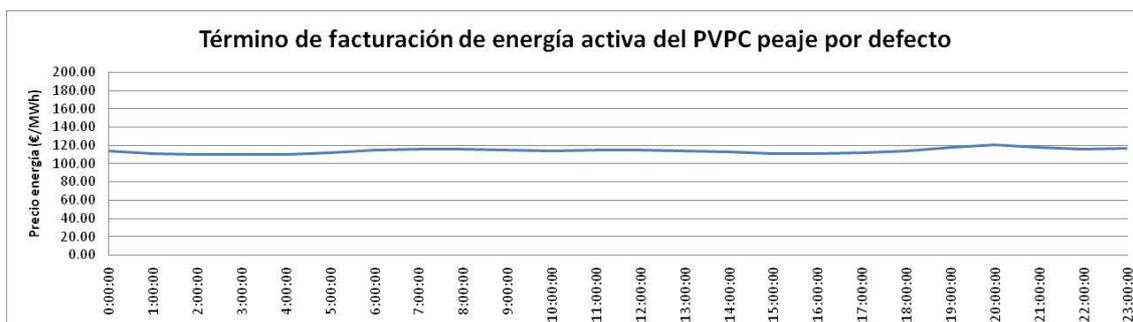


Figura 93. Término de facturación de energía activa del PVPC peaje por defecto: verano

Para la tarifa 2.0 DHA, tenemos un valor medio de:

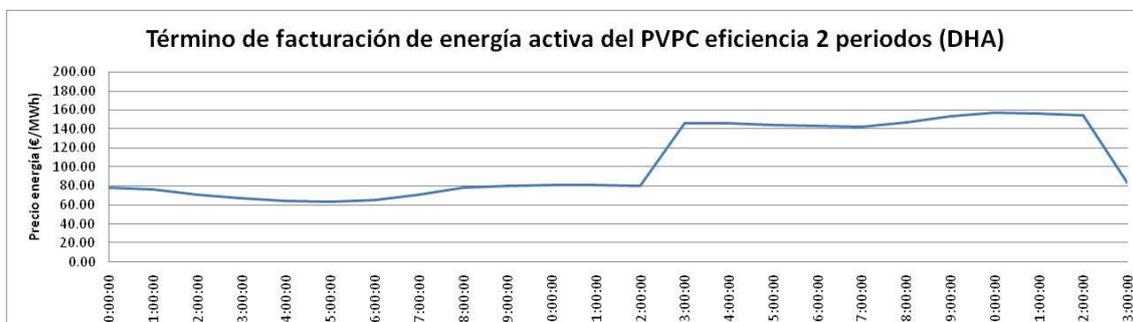


Figura 94. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA): invierno



Figura 95. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA): verano

Y para la tarifa 2.0 DHS:

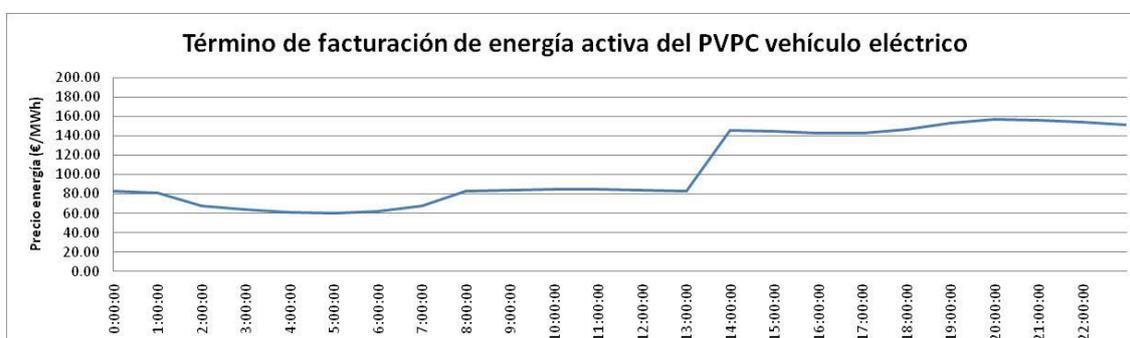


Figura 96. Término de facturación de energía activa del PVPC vehículo eléctrico: invierno

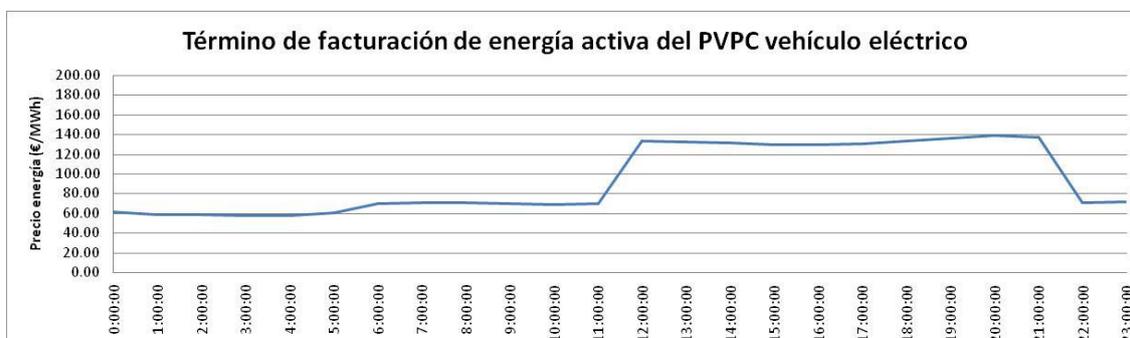


Figura 97. Término de facturación de energía activa del PVPC vehículo eléctrico: verano

Notar que, para el cálculo anterior, se ha asumido despreciable la estacionalidad con una periodicidad semanal. Esto se debe principalmente a la necesidad de simplificar el problema, y no tener diferentes precios para cada día de la semana (aunque se puede asumir el mismo desde miércoles a viernes). Con esto conseguimos simplificar el código de programación, sin cometer un gran error con respecto al precio real.

Una vez aclarado lo anterior, notar que los valores calculados de las desviaciones típicas, representan una dispersión de entorno a un 10 % frente al valor medio. Una forma visual de ver esto, es mediante los diagramas de *Box Whisker*, en los cuales se representa el valor medio, el valor máximo y mínimo que toma el precio para cada hora, y el cuartil 25% y 75%, donde el cuartil 25% es el valor del precio de la energía que deja por debajo un 25% del total de los datos, y el cuartil 75%, es el valor del precio de la energía que deja por debajo un 75%

del total de los datos. De esta forma, entre en cuartil 25% y el 75%, tenemos el 50% de los datos de la serie temporal. Gráficamente, tenemos para la tarifa por defecto:

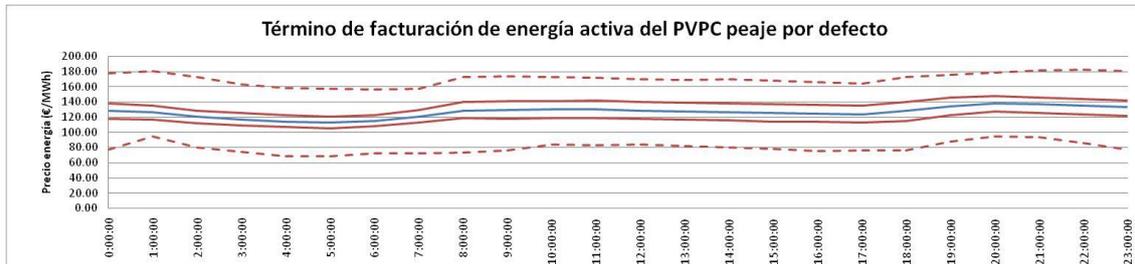


Figura 98. Término de facturación de energía activa del PVPC peaje por defecto: invierno

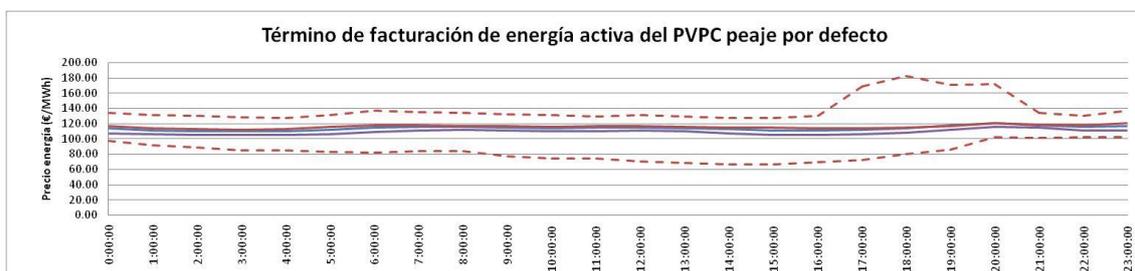


Figura 99. Término de facturación de energía activa del PVPC peaje por defecto: verano

Para la tarifa 2.0 DHA, tenemos:

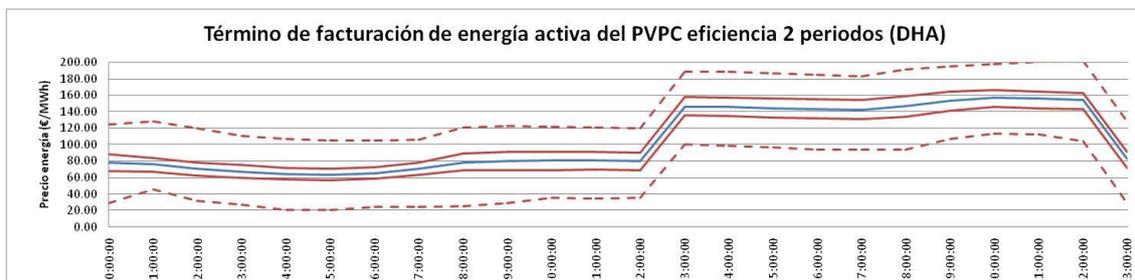


Figura 100. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA): invierno

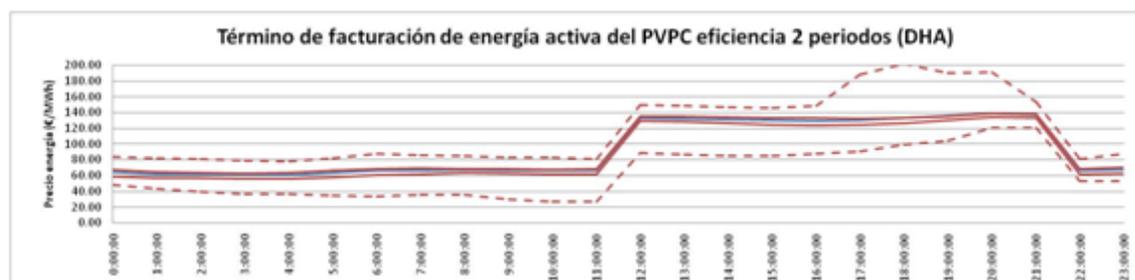


Figura 101. Término de facturación de energía activa del PVPC eficiencia 2 periodos (DHA): verano

Y para la tarifa 2.0 DHS:

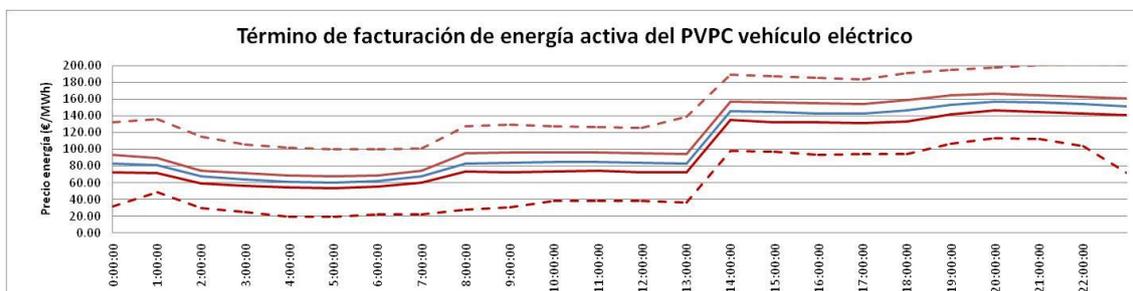


Figura 102. Término de facturación de energía activa del PVPC vehículo eléctrico: invierno

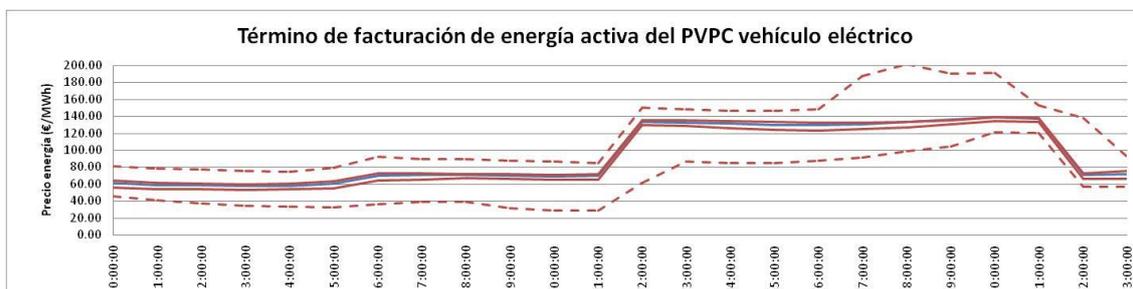


Figura 103. Término de facturación de energía activa del PVPC vehículo eléctrico: verano

Gráficamente, podemos concluir, que la mayor parte de los datos, se encuentran en torno al valor medio. Por lo que podemos asumir que el precio total de la energía a lo largo de la semana, y para cada estación y tarifa, es constante, teniendo un valor único del precio de la energía en la tarifa por defecto, y dos valores para el precio de la energía en las tarifas 2.0 DHA y 2.0 DHS. Notar que a pesar de que la tarifa 2.0 DHS es de 3 periodos, el precio durante el periodo supervalle es prácticamente el mismo que durante el periodo valle, por lo que se puede asumir un precio único, y tendrá un valor inferior al precio del periodo valle de la tarifa 2.0 DHA.

Así pues, finalmente, podemos definir lo siguientes precios de la energía (precio total de la energía):

Tabla 2. Precio de la energía para cada tarifa

€/MWh	Invierno		Verano	
	Punta	Valle	Punta	Valle
2.0 A	126,41		113,90	
2.0 DHA	148,89	74,19	133,18	65,82
2.0 DHS	149,44	73,94	133,15	65,51

Éstos serán los precios que se establecerán para que el usuario pueda obtener el coste de la energía asociado a su consumo eléctrico.

## 7. Programación

A lo largo del presente capítulo se detallará el código que ha sido programado para el funcionamiento del montaje. En primer lugar, se presentará la parte del código que pertenece al desarrollo en Arduino, que se corresponde con el microcontrolador y la comunicación de éste con el convertidor ADC, así como la realización de históricos y la publicación de los mismos. A continuación, se detallará el código en Android Studio que se corresponde con el envío de comandos y recepción de históricos, así como la representación de los mismos.

### 7.1. Arduino

Tal y como se ha introducido, a continuación, se describirá el código en Arduino. Éste lo podemos diferenciar en varias partes:

- Librerías utilizadas, las cuales contienen funcionalidades implementadas que nos serán de ayuda.

Nota: la librería utilizada para la comunicación entre el microcontrolador y el convertidor ADC, ha sido modificada para adaptarlo a un convertidor ADC con una resolución de 13 bits y, además, se ha implementado como una función (`leerADC_diferencial(canal)`), que se describirá en este capítulo en el punto *Operación de muestreo*, para así simplificar el programa.

- Declaración de variables globales.
- Desarrollo del programa, donde podemos distinguir las siguientes partes:
  - Captura de las señales de V y I, se tomarán 100 muestras por ciclo y cada 4 segundos.
  - Cálculo del valor eficaz de V y I, así como de la potencia activa, reactiva y aparente cada 4 segundos.
  - Guardado de los datos obtenidos en históricos. Se pretende tener históricos de todas las magnitudes minuto a minuto, hora a hora y día a día.
  - Conexión al servidor MQTT, lectura y publicación de mensajes en el topic suscrito.

#### 7.1.1. Librerías utilizadas y declaración de variables globales

Las librerías utilizadas son las siguientes:

```
#include<ESP8266WiFi.h>
```

Permite la conexión vías wifi del microcontrolador esp8266

```
#include<PubSubClient.h>
```

Permite conectarse a un servidor MQTT

```
#include<SPI.h>
```

Contiene funciones básicas para comenzar la comunicación SPI. No tiene implementado la forma en la que se envía los valores muestreados. El orden de transmisión se ha descrito en el capítulo 5.2.2.2 *Transmisión de datos*.

```
#include"time.h"
```

Contiene funciones para determinar la hora actual

```
#include"FS.h"
```

Permite la escritura y lectura en la memoria flash para la realización de históricos

Puesto que existen múltiples variables globales definidas, se explicarán las más importantes en los siguientes apartados para conocer las partes principales del programa.

Los códigos desarrollados en Arduino se estructuran en dos funciones: setup y loop. Mientras que el setup se ejecuta una sola vez al comienzo del programa, el loop se ejecuta de forma cíclica a continuación del void setup. Siguiendo esta estructura de programación se describirán las funciones principales del programa.

### 7.1.2. Función setup

Como ya se ha introducido, esta parte del programa se ejecuta una sola vez al inicio. Luego en esta parte del código se se lleva a cabo las siguientes funciones:

- Se inicializa la comunicación con el puerto serie.
- Se inicializa la comunicación SPI con el convertidor ADC
- Se inicializa el uso de la memoria flash del microcontrolador para guardar los históricos.
- Se establece la conexión wifi y con el servidor MQTT.
- Se obtiene la hora local mediante el servidor.
- Se Inicializan las variables
- Se establece el tiempo para comenzar la captura de datos con la hora local.

Siendo el código que realiza todas estas funciones el siguiente:

```
void setup() {
  //Iniciar comunicación con puerto serie
  Serial.begin(500000);
  //Iniciar uso de memoria flash
  SPIFFS.begin();
  //Iniciar conexión wifi (modificar ssid y password)
  setup_wifi();
  //Iniciar conexión con servidor MQTT
  client.setServer(mqtt_server, 23);
  client.setCallback(callback);
  //Configuración de la hora
  configTime(timezone, dst, "pool.ntp.org", "time.nist.gov");
  while (!time(nullptr))
  {
    delay(1000);
  }
  Serial.println("");
  //Inicializamos los vectores donde guardamos las nm muestras
  //obtenidas
  for(int i=0; i<nm; i++)
  {
    I[i] = 0;
    V[i] = 0;
  }
}
```

```

}
//Conexion SPI
pinMode(CS, OUTPUT);
digitalWrite(CS, HIGH);
SPI.begin();
SPI.setFrequency(4000000);
SPI.setBitOrder(MSBFIRST);
//Obtenemos el minuto posterior al leído mas 2 segundos
delay(5000);
time_t t_begin = time(nullptr);
t_muestreo = t_begin - (t_begin % 60) + 62;
}

```

Con el código anterior, el muestreo se realizará a partir de la hora `t_muestreo`, que será en el segundo dos del siguiente minuto, de forma que al tomar 15 muestras, el registro de minutos finalizará siempre en el segundo 58.

### 7.1.3. Función loop

Esta parte del programa se repite de forma cíclica tras la función setup. Para simplificar el código, se ha separado en dos estados diferentes:

- Estado 0: se llevan a cabo las operaciones de muestreo. Estas se dan cada 4 s a partir del segundo 2 de cada minuto, de forma que finaliza en el segundo 58 de cada minuto.

La operación de muestreo consiste en capturar 100 muestras con un paso de muestreo de 0,0002 s, de forma que con las 100 muestras se obtiene las formas de onda de la tensión y corriente para un periodo de la red de 20ms. Las 100 muestras se guardan en los vectores  $V[i]$  e  $I[i]$  y esta operación se repite cada 4 segundos, donde se comprueba además que el tiempo entre muestra y muestra sea constante. Una vez finalizan estas operaciones se pasa al estado 1

- Estado 1: con los vectores  $V[i]$  e  $I[i]$  se calcula el valor eficaz de tensión y corriente, así como valores máximos y mínimos. Asimismo, se calcula la potencia activa como el valor medio del producto de los valores instantáneos de tensión y corriente, la potencia aparente como el producto del valor eficaz de la tensión y corriente y la potencia reactiva a partir de las anteriores.

$$P = \frac{\sum V_i(t) \cdot I_i(t)}{nm} \quad (\text{LXIII.})$$

$$S = V_{ef} \cdot I_{ef} \quad (\text{LXIV.})$$

$$Q = \sqrt{S^2 - P^2} \quad (\text{LXV.})$$

A continuación esta información se guarda en la memoria flash y se comprueba si existe algún comando de petición de datos en el servidor. En caso de haber comandos publicados se envía la información solicitada. Cuando finalizan estas operaciones se pasa al estado 0.

7.1.3.1. Estado 0

En esta parte del programa tenemos que tomar 100 muestras cada 200  $\mu$ s (100 muestras por ciclo, es decir, 0,02s). Se leerá en modo diferencial en los canales 0 y 1 (configuración 0) y en los canales 2 y 3 (configuración 1), como se indica en la tabla del fabricante:

Control Bit Selections				Input Configuration	Channel Selection
Single /Diff	D2	D1	D0		
1	0	0	0	single ended	CH0
1	0	0	1	single ended	CH1
1	0	1	0	single ended	CH2
1	0	1	1	single ended	CH3
1	1	0	0	single ended	CH4
1	1	0	1	single ended	CH5
1	1	1	0	single ended	CH6
1	1	1	1	single ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

Figura 104. Configuración de bits MCP3304

El código para muestrear cada 200  $\mu$ s es el siguiente:

```

    if (estado==0)

//Operación de muestreo, cada 4 segundos muestrea nm muestras (100
//muestras)

    {

//Comenzamos la operacion de muestreo cuando el tiempo actual sea
//mayor o igual a t_muestreo

        while (t_muestreo <= time(nullptr))

    {

//Repetiremos la operación de muestreo si el At es mayor que un tiempo
//establecido (ver funcion Incr_T())

        do

    {

//Descartamos las 5 primeras muestras, lectura no estable en las
//primeras muestras

        for(int i=0; i<5; i++)
        {

            leerADC_Diferencial(0);
            leerADC_Diferencial(2);

        }

        time1 = micros();
        unsigned long tf = micros();
        unsigned long tact = micros();

//Muestreamos las nm muestras

        for(int i=0; i<nm; i++)
        {

            tf = tf + t_m;
            tiempo[i] = tact;

//Verificar canal de entrada diferencial (ver data sheet MCP3304)

            V[i] = leerADC_Diferencial(0);
            I[i] = leerADC_Diferencial(2);

//Muestreamos cada t_m us, es este caso, 200 us

            while (tf>tact)
            {

                yield();
                tact = micros();

            }

        } //Fin muestreo

        time2 = micros();
        time2 = time2 - time1;
        Freq = nm*1000000 / time2;

```

```

        estado = 1;
    }
    while (Incr_T(tiempo, nm)==1);
//Comprobamos que el At entre muestra y muestra no supere un cierto
//valor (ver Incr_T())

        t_muestreo += 4; //Muestreamos cada 4 s
    }
} //Fin operación de muestreo

```

Donde la función `Incr_T()` para descartar las muestras en el caso de que el paso de tiempo no sea constante es:

```

bool Incr_T(unsigned long tiempo[], int nm)
{
    unsignedlong At_max=0;
    unsignedlong At;
    for (int i=0;i<nm;i++)
    {
        if(i>1)
        {
            At=tiempo[i]-tiempo[i-1];
            if (At>At_max)
            {
                At_max=At;
            }
        }
    }
    if (At_max<=210)
    return 0;
    if (At_max>210)
    return 1;
}

```

Las 100 muestras se capturan mediante la función `leerADC_diferencial(canal)`:

```

short leerADC_Diferencial(short canal)
{
    digitalWrite(CS, LOW); //Se activa la lectura/escritura
    SPI.write(Diferencial | (canal >> 1));
    short recibido = SPI.transfer16(canal << 15);
    digitalWrite(CS, HIGH);
}

```

```

short dato = recibido & 0x0FFF;

if( recibido& 0x1000 )
{
    dato -= 4096;
}
return dato;
}

```

Justo es esta parte del código la que permite obtener de forma experimental la forma de onda durante un ciclo para verificar que no se excede el rango de entrada del convertidor ADC. Esto se vio en el *Capítulo 3. Diseño de los sensores de tensión y corriente*.

La función leerADC\_diferencial(canal) es la parte más importante en este punto puesto que se encarga de transmitir la información siguiendo el criterio que marca el fabricante del MCP3304 y que se vio en el *Capítulo 4. Captura digital*.

El orden de proceder es:

- En primer lugar, la señal de la línea CS se establece a nivel bajo para iniciar la comunicación SPI. En la función leerADC\_Diferencial(short canal) esto es:

```
digitalWrite(CS, LOW)
```

- A continuación, primero escucha el esclavo, indicándose que la señal a muestrear es en modo diferencial y el canal, que al ser en modo diferencial serán dos, aunque se codifica como uno (Figura 104). En la función leerADC\_Diferencial(short canal) esto es:

```
SPI.write(Diferencial | (canal >> 1))
```

- Después, el maestro escucha y recibe el valor muestreado en binario con una resolución de 12 bits más el de signo. En la función leerADC\_Diferencial(short canal) esto es:

```
short recibido = SPI.transfer16(canal << 15);
```

- Una vez recibido el valor muestreado en binario se vuelve a poner a nivel alto la líneaCS, esto es:

```
digitalWrite(CS, HIGH);
```

- Y, por último, se codifica a digital de 16 bits con signo el dato recibido:

```

short dato = recibido & 0x0FFF;
if( recibido& 0x1000 )
{
    dato -= 4096;
}
return dato;

```

Esta parte del código se ha adaptado de una librería publicada en la plataforma GitHub<sup>(25)</sup> que se ha implementado como una función por simplicidad.

### 7.1.3.2. Estado 1

Antes de proceder al cálculo de las magnitudes, se debe obtener el valor inicial de las señales. Si recordamos de los capítulos anteriores, para la tensión, se ha hecho uso de un divisor de tensión diferencial, y para la corriente, de un transformador de corriente, para adaptar los valores al rango de la instrumentación. Si recordamos esas expresiones:

$$V_{out1} - V_{out2} = (V_{in1} - V_{in2}) \cdot \frac{R_2}{R_1 + R_2} \quad (\text{LXVI.})$$

Donde  $V_{out1}$  y  $V_{out2}$  es la tensión que se obtiene entre las resistencias que forman el divisor de tensión diferencial, que se corresponde con la señal que llega al convertidor ADC, y  $V_{in1}$  y  $V_{in2}$  es la tensión de fase y neutro.

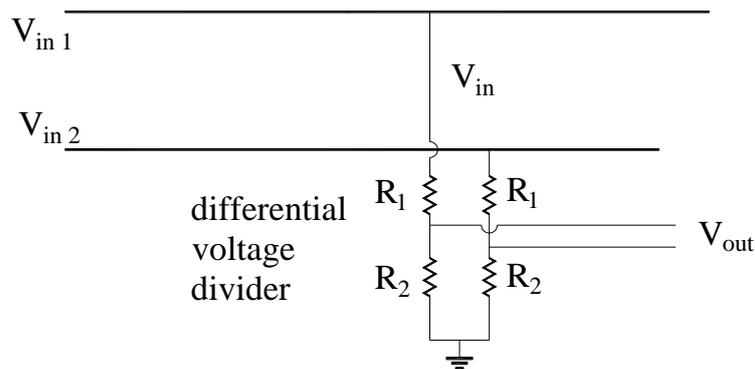


Figura 105. Divisor de tensión diferencial

Y para la corriente tenemos:

$$\frac{I_1}{I_2} = \frac{1}{r_t} \quad (\text{LXVII.})$$

Donde  $I_1$  será la corriente del primario, que es la realmente consumida por el usuario, y la corriente  $I_2$  es la del secundario. Recordamos que la relación de transformación era de 4000:1.

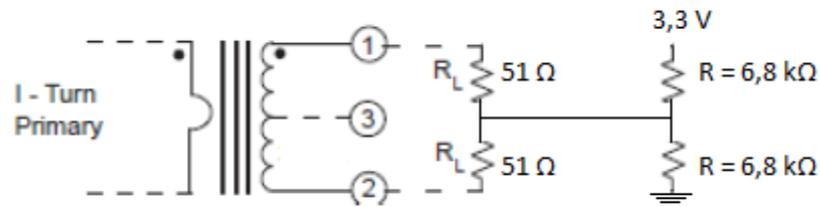


Figura 106. Esquema transformador de corriente

Ahora este cambio se tiene que deshacer para obtener así la señal original, que es el consumo del usuario, por lo que:

$$V_{in1} - V_{in2} = (V_{out1} - V_{out2}) \cdot \frac{R_1 + R_2}{R_2} \quad (\text{LXVIII.})$$

$$I_1 = \frac{1}{r_t} \cdot I_2 \quad (\text{LXIX.})$$

Además, tenemos que pasar del valor digital al valor en analógico. Para esto se debe tener en cuenta que el valor máximo en analógico que es 3,3V, se corresponde con el máximo valor en digital, que es 4095, por tener una resolución de 12 bits, por lo que finalmente quedaría:

$$V_{in1} - V_{in2} = (V_{out1} - V_{out2}) \cdot \frac{R_1 + R_2}{R_2} \cdot \frac{3,3}{4095} \quad (\text{LXX.})$$

$$I_1 = \frac{1}{r_t} \cdot I_2 \cdot \frac{3,3}{4095} \quad (\text{LXXI.})$$

Notar que, el valor medido por el convertidor es la caída de tensión que produce  $I_2$  en las dos resistencias  $R_L$ , por lo que la corriente en el secundario sería:

$$I_2 = (V_1 - V_2) \cdot \frac{1}{2R_L} \quad (\text{LXXII.})$$

Por lo que  $I_1$  será:

$$I_1 = \frac{1}{r_t} \cdot (V_1 - V_2) \cdot \frac{1}{2R_L} \cdot \frac{3,3}{4095} \quad (\text{LXXIII.})$$

Donde  $V_1 - V_2$  es la señal que se captura con el convertidor.

### Calibración de los sensores

Para el cálculo del valor eficaz, se tiene que calibrar los sensores. Para ello se cuenta con un multímetro.

Para calibrar la tensión, se realizarán múltiples lecturas del valor eficaz de la tensión con el código de programación descrito anteriormente, y se calibrará con el valor medio la tensión obtenida con el multímetro. De esta forma, podemos calcular la tensión como:

$$(V_{in1} - V_{in2}) = r_V \cdot (V_{in1} - V_{in2})_{medido\ con\ sensor} \quad (\text{LXXIV.})$$

Siendo  $r_V$ :

$$r_V = \frac{\bar{V}_{ef,medido\ con\ sensor\ de\ V}}{V_{ef,medido\ con\ multímetro}} \quad (\text{LXXV.})$$

De forma que se mide el valor eficaz cada minuto durante un número razonable de veces para tener un valor representativo, por ejemplo 15 valores de valor eficaz, por lo que el montaje tomará muestras durante 15 min y calcula el valor eficaz de cada minuto. La constante de proporcionalidad entre el valor medio obtenido del montaje y el valor eficaz medido con el multímetro será la constante de calibración.

Para calibrar la corriente, se utilizará una resistencia eléctrica (estufa eléctrica) como carga y se medirá el consumo con 1 y 2 resistencias en marcha. Tomados los valores obtenidos con el sensor de corriente y con el multímetro, se ajustan los puntos mediante una recta que pasa por 0 (para hacer coincidir que cuando no haya consumo, la corriente medida sea 0). La recta de ajuste se muestra a continuación:

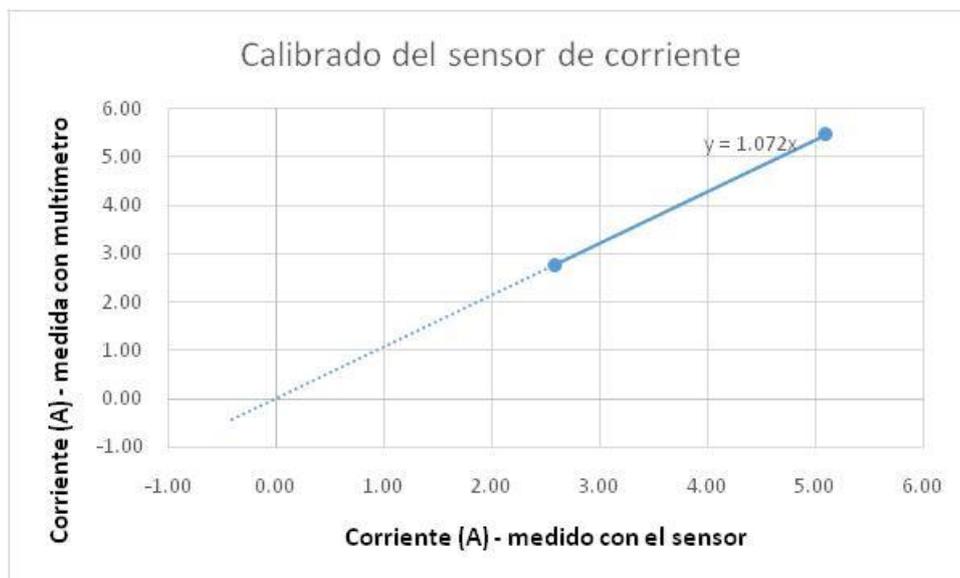


Figura 107. Calibrado del sensor de corriente

Siendo 1,072 la pendiente de la recta que permite obtener la corriente a partir del valor medido como:

$$I_{1,consumo} = 1,072 \cdot I_{1,medido\ con\ sensor\ I}$$

(LXXVI.)

Una vez obtenido el valor de las magnitudes de V y I, se procede al cálculo de la potencia activa (como el valor medio del producto del valor instantáneo de tensión y corriente), la potencia aparente (como el producto del valor eficaz de la tensión y corriente) y la potencia reactiva (a partir de las anteriores).

#### Guardado de los datos en históricos

La operación que se realiza en este punto, es la de guardar en variables locales las magnitudes ya calculadas. Tras 15 muestras separadas por 4 segundos, se completa 1 minuto. Tras 1 minuto muestreado, tenemos valores máximos, medios y mínimos de todas las magnitudes a lo largo de ese minuto. Una vez llegados a este punto, el valor guardado en las variables locales se guarda en la memoria flash en un histórico de minutos, que se irá llenando conforme se completa cada minuto.

Una vez completados 60 registros se tendrá 1 hora, por lo que se procede a añadir un registro en el histórico de horas, con el que, de forma análoga, una vez completado 24 registros, se añade uno nuevo en el histórico de días.

Es importante destacar que se puede dar el caso de desconexión del microcontrolador, por lo que en la programación se tiene que tener en cuenta esto, así como el llenado de los registros no obtenidos. Se ha optado por llenar los instantes no medidos por 0, puesto que se considera que si no se ha medido nada es debido a un corte en el suministro, por lo que el consumo es 0.

### Recepción y envío de mensajes al servidor MQTT

La forma de proceder para enviar y recibir comandos es la siguiente:

- Cada 4 segundos, una vez tomadas las 100 muestras, se comprueba el estado del servidor MQTT y si se ha publicado algún comando en el topic suscrito.
- Si se ha publicado algún comando o mensaje, éste se recibe y se guarda en una variable. La forma por la que se ha optado para publicar comandos por el dispositivo Android es el siguiente:
  - El usuario debe seleccionar la magnitud (V, I, P, Q, S y precio) y escala temporal (minutos, horas y días) que desee, los cuales son codificados de 0 a 5 y de 0 a 2 respectivamente.
  - Se publica un comando desde el dispositivo Android al servidor con la estructura RAB, siendo A un número del 0 al 5 y B del 0 al 2, en función de la magnitud y escala temporal seleccionada.

Si el mensaje publicado empieza por el carácter R (Report) seguido de dos caracteres es debido a que se ha solicitado datos de los históricos.

- Una vez conocido que se solicita algún dato del histórico se determina cual es, puesto que es conocida la magnitud y la escala temporal por los dos caracteres restantes.
- El mensaje de respuesta se prepara, comenzado por el carácter D (Data) seguido de otros tres caracteres que definen la magnitud de 0 a 5, la escala temporal de 0 a 3 y si es valor máximo, medio y mínimo de 3 a 0 respectivamente. Con esto definimos el contenido del mensaje de respuesta, y a continuación se escribe los datos leídos del histórico.
- Los datos del histórico son del tipo float, por lo que tienen un tamaño de 4 bytes. Para no tener mensajes demasiado largos, se procede a codificar los históricos de la siguiente forma:
  - Los valores de los históricos se codifican en variables de tipo short, que son de un tamaño de 2 bytes, esto es 16 bits. Para ello, tenemos que para cada una de las magnitudes existe un valor máximo, que será 400 V para la tensión 45 A para la corriente 10 kW para la potencia activa 1 kVAR para potencia reactiva y 10 KVA para la potencia aparente. Estos valores máximos se corresponden con un valor en digital de 32768 (16 bits), y así convertimos los valores de tipo float a short. Matemáticamente la operación que se lleva a cabo es:

$$\text{Dato magnitud } i_{short} = \text{Dato magnitud } i_{float} \cdot \frac{32768}{\text{Valor máximo magnitud } i}$$

(LXXVII.)

- A continuación, en el mensaje de respuesta se escribirán los valores en dos bytes
- Finalmente, debido a que existen limitaciones en el envío de mensajes cuando nos vamos a mensajes con un tamaño mayor a 100 bytes, se procederá a enviar los mensajes en distintos paquetes para no sobrepasar el límite.

El código simplificado es el siguiente:

```

    if (estado==1)
//Operación de mostrar datos por puerto serie, suscribir y publicar
//datos en servidor MQTT
    {
        unsigned long At_max = 0;
        unsigned long At = 0;
        bool MQTT_connect = 0;
//Abrimos puerto serie
        Serial.begin(500000);
//Comprobamos que estamos conectados al servidor MQTT una vez por cada
//muestra tomada
        if (time(nullptr) % 60 <= 50)
        {
            if (!client.connected())
            {
                MQTT_connect = reconnect();
            }
            else
            {
                MQTT_connect = 1;
            }

            if (MQTT_connect == 1)
            {
                client.loop();
            }
        }

//Obtenemos el valor de V e I a partir de los valores medidos en el
//convertidor
        for (int i=0;i<nm;i++)
        {
            V[i] = V[i]*rt_v*re;
            I[i] = I[i]*rt_i*re;
        }

        Irms = 0;
        Vrms = 0;
        S = 0;
        P = 0;
        Q = 0;

```

```

//Calculo del valor eficaz, potencia activa, reactiva y aparente
    for (int i=0;i<nm;i++)
    {
        Vrms = Vrms + V[i]*V[i];
        Irms = Irms + I[i]*I[i];
        P = P + V[i]*I[i];
    }
Vrms = Vrms / nm;
Irms = Irms / nm;
Vrms = pow(Vrms, 0.5);
Irms = pow(Irms, 0.5);
Vrms = Vrms * 218.00 / 185.77;
Irms = Irms * 1.0724;
P = P / nm;
S = Vrms*Irms;
Q = pow(S*S - P*P , 0.5);
//Fin calculo valor eficaz, potencia activa, reactiva y aparente
//Guardamos el valor medio máximo y mínimo de todas las magnitudes
//En este ejemplo sólo se determina para la tensión
//durante 1 min (15 //muestras espaciadas 4 s)
    if (pos_muestreo>0) //Actualizamos la matriz mr_m
    {
        mr_m [v_mag][val_med] = mr_m [v_mag][val_med] + Vrms;
//Maximo y minimo tension
    if (Vrms > mr_m [v_mag][val_max])
        mr_m [v_mag][val_max] = Vrms;
        if (Vrms < mr_m [v_mag][val_min])
            mr_m [v_mag][val_min] = Vrms;
        pos_muestreo++;
    }
    if (pos_muestreo==0)
//Inicializamos la matriz mr_m si pos_muestreo es 0
    {
        for (int i=0;i<3;i++)
        {
            mr_m [v_mag][i] = Vrms;
        }
        pos_muestreo++;
    }
}

```



```
//Reiniciamos contador pasado 1 min (15 paquetes de muestras
//espaciados 4 s)
if (((time(nullptr) % 60) >= 58) || (pos_muestreo>=15))
//Posibilidad de algun muestreo no realizado
{
    for(int i=0;i<nmag;i++)
    {
        mr_m [i][val_med] = mr_m [i][val_med] / pos_muestreo;
    }
    time_t min_muestra = time(nullptr);
    historico_min(min_muestra);
    pos_muestreo = 0;
}
//Final de guardado de los valores maximo, minimos y medios de la
//magnitudes en //mr[nmag][ns]
estado=0;
}
}
```

## 7.2. Android Studio

Para la realización del código en Android Studio, se ha hecho uso de un proyecto de ejemplo que implementaba exclusivamente la funcionalidad básica de comunicación MQTT ya existente publicado en la plataforma GitHub <sup>(26)</sup>. Sobre este proyecto, se han añadido las funcionalidades pertinentes para recibir los comandos de operador de la App a través de la interfaz de usuario (IU), enviar comandos al microcontrolador de lectura y registro vía MQTT según la codificación indicada en 7.1, recibir mensajes vía MQTT desde el microcontrolador de lectura y registro con la información solicitada, descodificarlos y representarlos datos repotados por el sistema de captura gráficamente como series temporales. El código, se estructura en las siguientes funciones:

- **OnCreate:** esta función se llama al iniciar la aplicación, y llama a las diferentes funciones para que crear la gráfica así como toda la interfaz de usuario e iniciar la comunicación con el servidor MQTT.
- **UiSetup:** esta función envía comandos al servidor MQTT en cuanto el usuario ha solicitado información sobre una cierta magnitud en una escala temporal, es decir, genera y envía los mensajes al microcontrolador de lectura y registro a partir de las acciones del operador sobre la interfaz de usuario.
- **MQTTstart:** establece conexión con el servidor MQTT.
- **GraphSetup:** esta función prepara las gráficas de series temporales que mostrarán los datos capturados al usuario: escalas, tamaños, colores, leyendas, etc.
- **UpdateGrapg:** actualiza la gráfica cuando se recibe mensajes de respuesta a los comandos del usuario.
- **OnDestroy:** Al cerrar la aplicación se desubscribe del servidor MQTT y se liberan los recursos utilizados

Estas funciones se describen con más detalle a continuación junto con el código simplificado.

### *Oncreate*

Esta función se ejecuta al iniciar la aplicación y hace la llamada a las distintas funciones para hacer funcionar el programa. El código es el siguiente:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // assign user interface resource to app
    setContentView(R.layout.activity_main);

    // allocate memory for captured data
    curvas = new double[5][tamcurva];
    //Crea la gráfica y la interfaz de usuario con botones "satelite"
    graphsetup();
    satmenusetup();
    //Envía comandos al servidor cuando el usuario lo requiere
    uisetup();
    //Establece la comunicación con el sevidor MQTT
    mqttstart();
}
```

## GraphSetup

En esta función crea el área de la gráfica así como las series que se representan. El código se muestra a continuación:

```
private void graphsetup() {
    dataset = new XYMultipleSeriesDataset();
    renderer = new XYMultipleSeriesRenderer();
    serie_max = new XYSeries("Valor máximo");
    serie_med = new XYSeries("Valor medio");
    serie_min = new XYSeries("Valor mínimo");
    dataset.addSeries(serie_max);
    dataset.addSeries(serie_med);
    dataset.addSeries(serie_min);
    serie_max_r = new XYSeriesRenderer();
    serie_med_r = new XYSeriesRenderer();
    serie_min_r = new XYSeriesRenderer();
    renderer.addSeriesRenderer(serie_max_r);
    renderer.addSeriesRenderer(serie_med_r);
    renderer.addSeriesRenderer(serie_min_r);
    renderer.setPanEnabled(false, false);
    renderer.setZoomEnabled(false, false);

    LinearLayout layout = (LinearLayout) findViewById(R.id.uilayouts);
    gview = ChartFactory.getLineChartView(this, dataset, renderer);
    layout.addView(gview,
        new LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
            LinearLayout.LayoutParams.WRAP_CONTENT));

    renderer.setMargins(new int[]{40, 60, 40, 20});
    renderer.setMarginsColor(Color.WHITE);
    renderer.setAxesColor(Color.BLACK);
    renderer.setXLabelsColor(Color.BLACK);
    renderer.setYLabelsColor(0, Color.BLACK);
    renderer.setYLabelsAlign(Paint.Align.RIGHT, 0);
    renderer.setLabelsTextSize(20);
    renderer.setXTitle("time (min)");
    renderer.setYTitle("Volts");
    renderer.setAxisTitleTextSize(20);
    renderer.setLabelsColor(Color.BLACK);
    renderer.setYAxisMax(4.0);
    renderer.setYAxisMin(-4.0);
    renderer.setGridColor(Color.DKGRAY);
    renderer.setShowGrid(true);
    renderer.setLegendTextSize(20);
    serie_max_r.setColor(Color.RED);
    serie_med_r.setColor(Color.BLUE);
    serie_min_r.setColor(Color.GREEN);

    gview.repaint();
}
```

## UISetup

Envía un comando al servidor MQTT cuando el usuario solicita información sobre una magnitud en una determinada escala temporal.

```
private void uisetup() {

    button1P = (Button) findViewById(R.id.button1P);
    button2P = (Button) findViewById(R.id.button2P);
    button1P.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (Pselect != 0 || firstbuttonP) {
                firstbuttonP = false;
                Pselect = 0;
                button1P.setTextColor(Color.RED);
                button2P.setTextColor(Color.BLACK);
            }
        }
    });
    button2P.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (Pselect != 1) {
                firstbuttonP = false;
                Pselect = 1;
                button1P.setTextColor(Color.BLACK);
                button2P.setTextColor(Color.RED);
            }
        }
    });
}
```

## MQTTstart

Establece comunicación con el servidor MQTT y comprueba si se ha publicado algún mensaje. En caso de publicarse comprueba si el mensaje comienza por el carácter D, de ser así el mensaje se ha enviado por el microcontrolador y contiene la serie de datos que el usuario ha demandado, por lo que se llama a la función `updateGraph(message)` con el mensaje recibido.

```
protected void mqttstart() {
    pahoMqttClient = new PahoMqttClient();
    client = pahoMqttClient.getMqttClient(getApplicationContext(),
        "tcp://mqttnameserver", "androidkt");
    handler = new Handler() {
        @Override
        public void handleMessage(android.os.Message msg)
        {
            if (msg.what == 100)
            {
                byte[] message = (byte[])msg.obj;
                byte code = (byte)((byte)message[0] - (byte)'C');
                byte magnitud;
                if (message.length > 1)
                    magnitud = (byte)((byte)message[1] - (byte)'0');
                else
                    magnitud = 0;
                if (code == 0) {
```

```

} else if (code == 1)
    updateGraph(message);

    }
else if (msg.what == 101) {
}
    }
};

```

### UpdateGraph

Actualiza la gráfica con el mensaje recibido de la función `mqttstart()`.

Este mensaje contiene la magnitud y la escala temporal seleccionada por el usuario, pero antes de representarlo se debe acondicionar la información puesto que esta se envía codificada. En los siguientes extractos de código se resaltan los aspectos más importantes.

```

protected void updateGraph(byte[] message) {
mag_selec_txt = (TextView) findViewById(R.id.Mag_selec);
dtemp_selec_txt = (TextView) findViewById(R.id.DTemp_selec);
int length = message.length;
if (c_msg == 0)
{
serie_max.clear();
serie_med.clear();
serie_min.clear();
byte d_time = message[1];
switch (d_time)

```

Actualiza el eje de abcisas según la escala temporal seleccionada y el número de datos recibido

```

byte magnitud = message[2];
switch (magnitud)

```

Actualiza el eje de ordenadas según la magnitud seleccionada, descodifica los datos y ordena los mensajes para tener la serie de datos de más antiguo al más actual. Detacar el proceso de descodificación, que se describe a continuación:

```

short datashort;
byte high = message[position++];
    datashort = (short)high;
    datashort <<= 8;

if (high < 0)
    datashort |= 0x8000;
byte low = message[position];
    datashort += (short)low + (short)(low < 0 ? 256 : 0);

double data = datashort * (valor_max_mag) / 32768.0;

```

El proceso de descodificación consiste en ir leyendo el mensaje recibido en binario. Puesto que los datos ocupan 2bytes la información se lee de dos en dos, y a continuación se divide entre el factor usado para codificar descrito en (LXXVII.)

```

}

}

```

```
gview.repaint();
}
```

Por otro parte, cabe destacar cómo se ha calculado el precio de la energía eléctrica. Éste se calcula de la siguiente forma:

$$\begin{aligned}
 & \text{Coste de la electricidad (€)} \\
 &= \text{Consumo eléctrico (kW)} \cdot \text{Precio energía eléctrica} \left( \frac{\text{€}}{\text{kWh}} \right) \\
 & \cdot \text{Densidad temporal seleccionada (min)} \cdot \frac{1 \text{ h}}{60 \text{ min}}
 \end{aligned}$$

(LXXVIII.)

Donde la electricidad consumida es el dato recibido desde el microcontrolador, y el precio de la energía se obtiene a partir del Sistema de Información del Operador del Sistema (eSios)<sup>(21)</sup>, donde se publica el coste de la energía para consumidores domésticos que están sujetos a las tarifas del PVPC. Podemos identificar dos tipos de tarificación que son los más usados.

- Peaje por defecto (2.0 A): en este tipo de tarificación no existe discriminación horaria.
- Eficiencia 2 periodos (2.0 DHA): en este caso sí existe discriminación horaria.

Adicionalmente, hay una tarifa supervalle, pensado para el vehículo eléctrico, pero no se ha considerado para este proyecto.

Notar que, en la expresión anterior, el precio de la energía que se tiene como dato, es el coste asociado al consumo de 1 kW de potencia durante una hora, es por eso por lo que en función de la densidad temporal seleccionada, habrá que determinar el coste asociado al consumo de 1 kW durante 1 min (si se elige como densidad temporal minutos), durante 60 min (si se elige como densidad temporal horas) o durante 1440 min (si se elige como densidad temporal días). De forma que para cada caso queda:

$$\begin{aligned}
 & \text{Coste de la electricidad histórico minutos (€)} \\
 &= \text{Consumo eléctrico (kW)} \cdot \text{Precio energía eléctrica} \left( \frac{\text{€}}{\text{kWh}} \right) \cdot 1 \text{ min} \cdot \frac{1 \text{ h}}{60 \text{ min}}
 \end{aligned}$$

(LXXIX.)

$$\begin{aligned}
 & \text{Coste de la electricidad histórico horas (€)} \\
 &= \text{Consumo eléctrico (kW)} \cdot \text{Precio energía eléctrica} \left( \frac{\text{€}}{\text{kWh}} \right) \cdot 60 \text{ min} \cdot \frac{1 \text{ h}}{60 \text{ min}}
 \end{aligned}$$

(LXXX.)

$$\begin{aligned}
 & \text{Coste de la electricidad histórico días (€)} \\
 &= \text{Consumo eléctrico (kW)} \cdot \text{Precio energía eléctrica} \left( \frac{\text{€}}{\text{kWh}} \right) \cdot 1440 \text{ min} \\
 & \cdot \frac{1 \text{ h}}{60 \text{ min}}
 \end{aligned}$$

(LXXXI.)

Estas expresiones se implementarán en el código para calcular el coste de la electricidad, con los precios publicados en el Sistema de Información del Operador del Sistema.

Nota: para realizar este cálculo, se considera que en función de la densidad temporal, se consume una potencia constante a lo largo de 1 minuto, 1 hora o 1 día

### InitSubscribe

Se establece y se comprueba la conexión con el servidor MQTT

```
protected void initsubscribe() {
    if(!subscribed) {
        try {
            pahoMqttClient.subscribe(client, "MQTT Examples", 1);
        } catch (MqttException e) {
            e.printStackTrace();
        }
    }
    else
        inittimer.cancel();
    try {
        pahoMqttClient.publishMessage(client, "C00", 1, "MQTT Examples");
    } catch (MqttException e) {
        e.printStackTrace();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}
}
```

### OnDestroy

Cuando se cierra la aplicación se desubscribe del servidor MQTT

```
protected void onDestroy() {
    // stopping the timer
    if(scantimer != null)
        scantimer.cancel();
    // disconnecting & releasing used BTH resources
        //dd.disconnect();
        //ddm.releaseDaqDevice(dd);

        // unsubscribe from Mqtt server
    if(subscribed) {
        try {
            pahoMqttClient.unsubscribe(client, "MQTT examples");
        } catch (MqttException e) {
            e.printStackTrace();
        }
    }
    super.onDestroy();
}
}
```

### 7.2.1. Manual de usuario

Finalmente, y para terminar con este capítulo, se mostrará cómo se representan los datos y una guía para que el usuario se pueda manejar.

El uso de la aplicación es muy sencillo, tal y como se aprecia en la siguiente imagen, en la parte inferior de la pantalla se tienen dos botones (menús satélite, que reciben esta denominación porque, al pulsarlos, despliegan una serie de botones dispuestos alrededor del botón original con las opciones disponibles), uno en cada esquina. El inferior izquierdo para seleccionar la densidad temporal, y el inferior derecho para seleccionar la magnitud.



Figura 108. Interfaz de usuario de la aplicación

Así pues, pulsando el botón inferior izquierdo de selección de escala temporal, se despliegan las diferentes densidades temporales con las que se puede mostrar cada magnitud. Estas son: minutos, horas y días.



Figura 109. Selección de la escala temporal

De forma análoga, pulsando el botón inferior derecho para la selección de la magnitud, se despliega las diferentes magnitudes que se pueden mostrar al usuario, estas son: tensión, corriente, potencia activa, reactiva, aparente y una sexta magnitud que sería el precio de la energía.



Figura 110. Selección de la magnitud

Adicionalmente en la parte superior se cuenta con dos botones para la selección de la tarifa, los precios para cada tarifa, así como los períodos tarifarios donde aplica cada precio, se establece en *6Determinación del precio de la energía*.

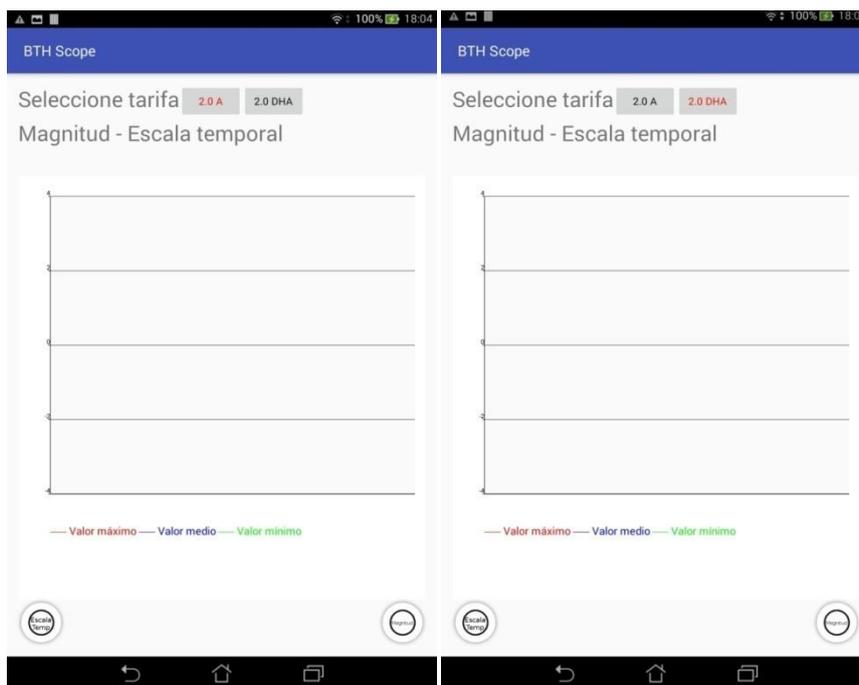


Figura 111. Selección de la tarifa eléctrica

Adicionalmente, a modo informativo se muestra la magnitud y la escala temporal que se representa en la gráfica.

Para la representación de la gráfica, como se ve en Figura 112, en el eje de abscisas se representa el tiempo de los últimos minutos, horas o días, según la escala temporal seleccionada, siendo los primeros valores de la serie temporal los datos más antiguos registrados.

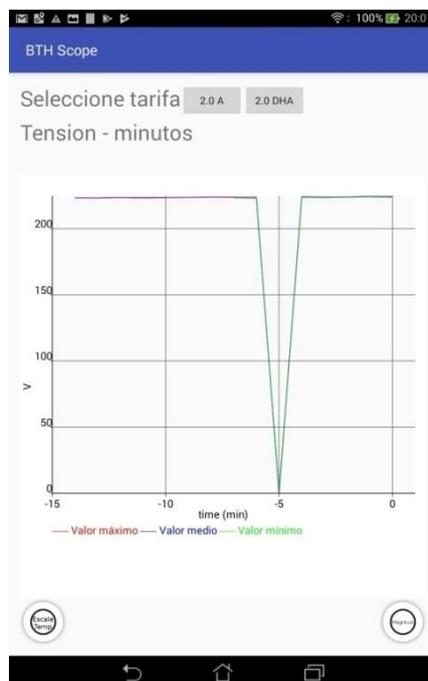


Figura 112. Representación de los datos

Notar cómo en estos ensayos tenemos varios valores en cero, esto es debido a que durante los ensayos la señal wifi se encontraba lejos del microcontrolador. Así pues, teniendo en cuenta que la antena para conectarse a la señal wifi del microcontrolador es pequeña, esto nos limita para la conexión y la recepción y envío de mensajes. Es por esto por lo que tenemos huecos donde no hemos registrado datos.

En cualquier caso, se puede apreciar la funcionalidad de la aplicación, para poder observar a distintas densidades temporales (minutos, horas y días), diferentes magnitudes (V, I, P, Q, S y precio de la energía consumida), el valor máximo, medio y mínimo, pudiendo tener un control de la calidad de onda de la tensión y teniendo un control de la potencia consumida, así como al coste asociado al mismo.

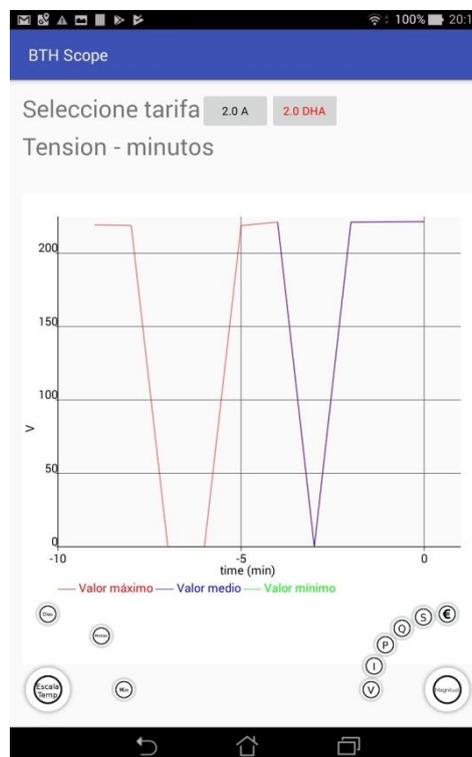


Figura 113. Registro de tensión en minutos

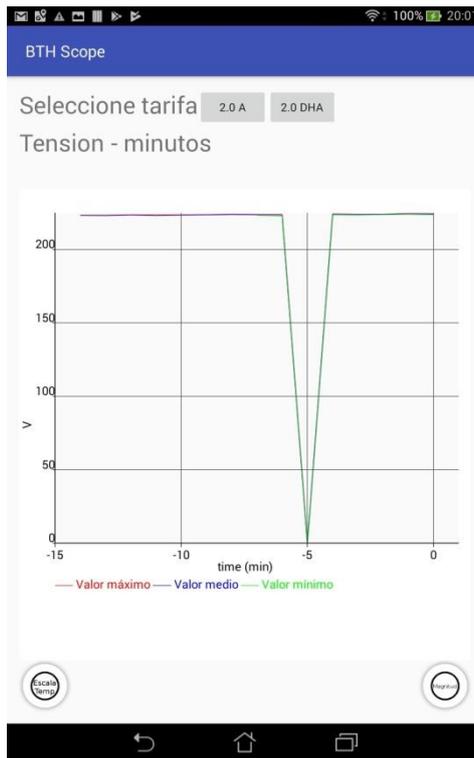


Figura 114. Registro de tensión en minutos

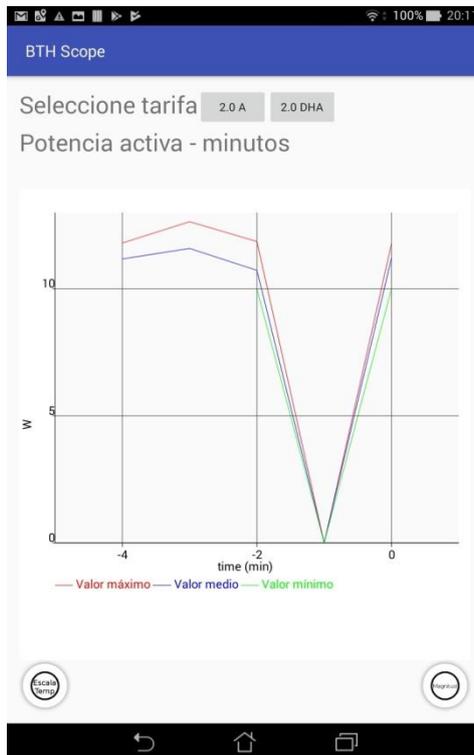


Figura 115. Registro de potencia activa en minutos

## 8. Conclusiones

Como conclusiones, y en primer lugar, cabe destacar la importancia de haber desarrollado e implementado a partir de elementos de coste reducido un sistema capaz de monitorizar el consumo eléctrico, no sólo en una vivienda unifamiliar, sino también en cualquier planta o instalación industrial, pues esta es la forma de poder identificar consumos anómalos derivados de un mal uso o por falta de conciencia de los usuarios.

Asimismo, notar cómo en el capítulo de desarrollo de los sensores, el hecho de encontrarnos en una industria no implica ninguna restricción, únicamente habría que cambiar el fondo de escala, modificando la tensión de alimentación, si fuera necesario, y la corriente máxima en las ecuaciones.

De esta forma, resulta de gran utilidad poder contar con este tipo de dispositivos, sobre todo en el caso de implantar mejoras de eficiencia energética, puesto que sería posible monitorizar el consumo antes y después de la implantación de las medidas y poder verificar que se está produciendo unos ahorros sostenidos en el tiempo gracias a esta mejora.

Por otra parte, un punto importante de este proyecto y que ha marcado el diseño del dispositivo es el coste de fabricación, en torno a 40 €. Esto unido a las funcionalidades que presenta lo convierte en una alternativa muy atractiva.

Finalmente, remarcar de nuevo la importancia de poder monitorizar una instalación, pues gracias a esto es posible llevar un control correcto, con la posibilidad de mejorar el uso de la energía.

## Planteamiento del presupuesto

En el presente capítulo se llevará a cabo el análisis del coste final de la solución adoptada. Para la implementación del equipo en la vivienda, el cliente (en este caso un consumidor doméstico) tendrá el coste asociado por un lado al propio equipo, y por el otro lado, de la mano de obra requerida para instalar el equipo.

### Mano de obra

Para la elaboración del presente proyecto, ha intervenido un ingeniero graduado en ingeniería de la energía. Para llevar a cabo la instalación del montaje, el mismo ingeniero podría llevar a cabo esta labor, aunque de forma opcional, esta labor la podría realizar un técnico electricista. Con esto, y tomando como referencia los salarios brutos anuales de Tabla 3, y con el calendario de trabajo indicado en Tabla 4, se obtiene el salario bruto por hora:

Tabla 3. Precios de mano de obra

Trabajador	Salario bruto anual (€)	Salario bruto por hora (€/h)
Graduado en Ingeniería de la Energía	35 000	19,88
Técnico electricista	20 000	11,36

Tabla 4. Calendario de trabajo

<b>Días laborales anuales</b>	220
<b>Jornada laboral en horas</b>	8
<b>Horas de trabajo al año</b>	1760

## Materiales

Los materiales listados en Tabla 6y Tabla 6son aquellos necesarios para el montaje del equipo de captura de consumo eléctrico, así como para el desempeño del trabajo del ingeniero durante el trabajo de desarrollo, motivo por el cual, se considera el coste de un ordenador, donde se ha considerado una vida útil de 5 años y 1760 horas al año para obtener el coste horario. Notar que no se tiene en cuenta el coste de ningún software, puesto que los programas utilizados (Arduino y Android Studio) son libres y gratuitos, por lo que no se tiene ningún coste en licencias.

Tabla 5. Precios de materiales

Material	Unidades	Coste unitario(€)	Coste total(€)
Resistencia 301 kΩ	2	0,039 <sup>(27)</sup>	0,078
Resistencia 1,3 kΩ	2	0,068 <sup>(28)</sup>	0,136
Resistencia 4,32 kΩ	2	0,039 <sup>(29)</sup>	0,78
Resistencia 1 kΩ	2	0,064 <sup>(30)</sup>	0,128
Condensador 220 μF	2	0,107 <sup>(31)</sup>	0,214
Esp8266	1	7,69 <sup>(32)</sup>	7,69
MCP3308	1	4,185 <sup>(33)</sup>	4,185
Cable Unipolar <sup>1</sup>	1	20,57 <sup>(34)</sup>	0,2057 <sup>2</sup>

Tabla 6. Precios de materiales

Material	Unidades	Coste unitario(€)	Coste horario(€/h)
Ordenador	1	1000	0,11

De esta forma, de la Tabla 6, se tiene que el coste unitario derivado de los materiales para llevar a cabo el montaje, es de 13,41 €.

Destacar que los precios unitarios indicados en la tabla, vienen definidos en paquetes de varias unidades (ver referencias). En el caso de adquirir paquetes con mayor número de unidades se podría reducir el coste aún más. En cualquier caso este coste es notablemente inferior al de las alternativas que se pueden encontrar en el mercado, donde el precio es un orden de magnitud superior (desde los 100 €, como se vio en *Introducción*).

<sup>1</sup> Carrete de 100 m

<sup>2</sup> Se toma 1 m de cable para el montaje

## Unidades de obra. Precios descompuestos

Para la elaboración del presupuesto ha tenido en cuenta las unidades de obra detalladas a continuación.

Tabla 7. Unidad de obra nº 1

Nº Orden	Descripción de la unidad de obra		
U.O.1	Ud. ESTUDIO DEL ESTADO DEL ARTE		
	Investigación sobre las alternativas existentes en el mercado y análisis de las soluciones técnicas existentes para medir el consumo doméstico.		
	<b>COSTES DIRECTOS</b>		
Rendimiento	Descripción	Precio (€/h)	Importe (€)
	30 h. Graduado en Ingeniería Energética	19,88	596,4
	30 h. Ordenador	0,11	3,3
	2% Coste directo complementario		11,99
TOTAL PRECIO UNIDAD DE OBRA			611,7

Tabla 8. Unidad de obra nº 2

Nº Orden	Descripción de la unidad de obra		
U.O.2	Ud. ANÁLISIS Y SELECCIÓN DE LA SOLUCIÓN ADOPTADA		
	Análisis de las posibles opciones para llevar a cabo un equipo de reporte y captura de datos. Selección de la alternativa y de la instrumentación necesaria para llevarlo a cabo.		
	<b>COSTES DIRECTOS</b>		
Rendimiento	Descripción	Precio (€/h)	Importe (€)
	60 h. Graduado en Ingeniería Energética	19,88	1.192,8
	60 h. Ordenador	0,11	6,6
	2% Coste directo complementario		23,99
TOTAL PRECIO UNIDAD DE OBRA			1.223,39

Tabla 9. Unidad de obra nº 3

Nº Orden	Descripción de la unidad de obra		
U.O.3	Ud. DESARROLLO CÓDIGO DEL MICROCONTROLADOR		
	Desarrollo del código en Arduino implementando las funcionalidades para obtener y reportar el consumo doméstico		
<b>COSTES DIRECTOS</b>			
Rendimiento	Descripción	Precio (€/h)	Importe (€)
300	h. Graduado en Ingeniería Energética	19,88	5.997
300	h. Ordenador	0,11	33
	2% Coste directo complementario		119,94
TOTAL PRECIO UNIDAD DE OBRA			6.116,94

Tabla 10. Unidad de obra nº 4

Nº Orden	Descripción de la unidad de obra		
U.O.4	Ud. DESARROLLO DEL HARDWARE		
	Integración del convertidor ADC con el microcontrolador		
<b>COSTES DIRECTOS</b>			
Rendimiento	Descripción	Precio (€/h)	Importe (€)
60	h. Graduado en Ingeniería Energética	19,88	1.192,8
60	h. Ordenador	0,11	6,6
	2% Coste directo complementario		23,99
TOTAL PRECIO UNIDAD DE OBRA			1.223,39

Tabla 11. Unidad de obra nº 5

Nº Orden	Descripción de la unidad de obra		
U.O.5	Ud. DESARROLLO CÓDIGO DEL DISPOSITIVO ANDROID		
	Desarrollo del código en Android Studio para obtener el consumo doméstico que ha sido reportado por el microcontrolador y mostrarlo al usuario		
<b>COSTES DIRECTOS</b>			
Rendimiento	Descripción	Precio (€/h)	Importe (€)
300	h. Graduado en Ingeniería Energética	19,88	5.997
300	h. Ordenador	0,11	33
	2% Coste directo complementario		119,94
TOTAL PRECIO UNIDAD DE OBRA			6.116,94

Tabla 12. Unidad de obra nº 6

Nº Orden	Descripción de la unidad de obra		
U.O.6	Ud. MATERIALES		
	Materiales necesarios para realizar el montaje		
<b>COSTES DIRECTOS</b>			
	Descripción	Precio (€/ud)	Importe (€)
	Materiales	13,41	13,41
	2% Coste directo complementario		0,27
TOTAL PRECIO UNIDAD DE OBRA			13,68

Tabla 13. Unidad de obra nº 7

Nº Orden	Descripción de la unidad de obra		
U.O.7	Ud. INSTALACIÓN DEL EQUIPO		
	Instalación del equipo en la vivienda del cliente		
	<b>COSTES DIRECTOS</b>		
Rendimiento	Descripción	Precio (€/h)	Importe (€)
	1 h. Técnico electricista	11,36	11,36
	2% Coste directo complementario		0,22
TOTAL PRECIO UNIDAD DE OBRA			11,58

## Unidades de obra. Precios unitarios

U.O.1	Ud. ESTUDIO DEL ESTADO DEL ARTE	611,7€
U.O.2	Ud. ANÁLISIS Y SELECCIÓN DE LA SOLUCIÓN ADOPTADA	1.223,39 €
U.O.3	Ud. DESARROLLO CÓDIGO DEL MICROCONTROLADOR	6.116,94 €
U.O.4	Ud. DESARROLLO DEL HARDWARE	1.223,39 €
U.O.5	Ud. DESARROLLO CÓDIGO DEL DISPOSITIVO ANDROID	6.116,94€
U.O.6	Ud. MATERIALES	13,68 €
U.O.7	Ud. INSTALACIÓN DEL EQUIPO	11,58 €

## Presupuesto

Definidas las unidades de obra del proyecto, para llevarlo a cabo, el capital a invertir sería:

Estudio del estado del arte	611,7 €
Análisis y selección de la solución adoptada	1.223,39 €
Desarrollo código del microcontrolador	6.116,94 €
Desarrollo del hardware	1.223,39 €
Desarrollo código del dispositivo Android	6.116,94 €

<b>Presupuesto de Ejecución</b>	<b>15.292,36€</b>
Gastos Generales (15 %)	2.293,85€
Beneficio Industrial (6 %)	917,54€

<b>Presupuesto sin I.V.A</b>	<b>18.503,75€</b>
I.V.A. 21%	3.885,79€

<b>Presupuesto Final</b>	<b>22.389,54€</b>
--------------------------	-------------------

El presupuesto final asciende a:

VEINTIDOS MIL TRESCIENTOS OCHENTA Y NUEVE CON CINCUENTA Y CUATRO

Mientras que, el coste de cada equipo sería:

Materiales	13,68 €
Instalación del equipo	11,58 €
<hr/>	
<b>Presupuesto de Ejecución</b>	<b>25,26 €</b>
Gastos Generales (15 %)	3,79 €
Beneficio Industrial (6 %)	1,51 €
<hr/>	
<b>Presupuesto sin I.V.A</b>	<b>30,56€</b>
I.V.A. 21 %	6,42 €
<hr/>	
<b>Presupuesto Final</b>	<b>36,98 €</b>

El coste de cada equipo sería de:

TREINTA Y SEIS EUROS CON NOVENTA Y OCHO

Notar que el coste de cada equipo es de en torno a 40 €, sin embargo, desde un punto de vista económico, el precio final tiene que ser tal que en el largo plazo, cubra el coste de la inversión inicial de 22.389,54 €

## Bibliografía

1. Smappee. [En línea] [https://www.smappee.com/eu\\_es/home](https://www.smappee.com/eu_es/home).
2. Organización del consumidor. [En línea] <https://www.ocu.org/vivienda-y-energia/gas-luz/noticias/cuanta-energia-consume-una-casa-571584>.
3. Smappee. [En línea] [https://www.smappee.com/eu\\_es/eshop/monitors.html](https://www.smappee.com/eu_es/eshop/monitors.html).
4. **García, Miguel Ángel Pérez.** *Instrumentación electrónica*. Madrid : Paraninfo, 2014.
5. **Blum, Jeremy.** *Exploring Arduino: Tools and Techniques for Engineering Wizardry*. s.l. : Wiley.
6. Electrónica en general, Tipología de la señal. [En línea] [Citado el: 24 de 06 de 2018.] <http://electronicaengeneral.com/?tag=single-ended>.
7. Lecciones de Electrónica. [En línea] [Citado el: 24 de 06 de 2018.] <http://www.ifent.org/lecciones/lecciones.asp>.
8. Convertidores A/D y D/A. [En línea] [Citado el: 04 de 06 de 2018.] [http://www.geocities.ws/pnavar2/convert/adc/adc\\_31.html](http://www.geocities.ws/pnavar2/convert/adc/adc_31.html).
9. Módulos analógicos de entrada y salida. [En línea] [Citado el: 04 de 06 de 2018.] <http://controlreal.com/es/modulos-analogicos-de-entrada-y-salida/>.
10. Instrumentación y monitorización electrónica.
11. RS Components, transformador de medida de tensión. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/transformadores-de-pcb/7320528/?sra=pspk>.
12. RS Components, sensor de corriente efecto Hall. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/transductores-de-corriente/3019495/>.
13. RS Components, sensor de corriente transformador de corriente. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/transformadores-de-corriente/3997339/>.
14. Boletín Oficial del Estado: Real Decreto 216/2014. 2014.
15. PlayByte. [En línea] 2018. <http://www.playbyte.es/electronica/arduino/esp-01s-modulo-wifi-basado-en-esp8266/>.
16. Arduino. [En línea] 2018. <http://arduinoamute.blogspot.com.es/2016/12/nodemcu-dev-kit-v3-de-lolin.html>.
17. **Microchip.** Farnell. [En línea] <http://www.farnell.com/datasheets/1486116.pdf>.
18. Esp8266 Learning. [En línea] <http://www.esp8266learning.com/wemos-mcp3008-example.php>.
19. GitHub. [En línea] <https://github.com/nodesign/MCP3008>.
20. GitHub. [En línea] [https://github.com/Sillern/esp\\_mcp3008](https://github.com/Sillern/esp_mcp3008).
21. Sistema de Información del Operador del Sistema. [En línea] [https://www.esios.ree.es/es/analisis/10229?vis=1&start\\_date=21-04-](https://www.esios.ree.es/es/analisis/10229?vis=1&start_date=21-04-)

2018T00%3A00&end\_date=21-04-2018T23%3A00&compare\_start\_date=20-04-2018T00%3A00&groupby=hour&compare\_indicators=10230,10231.

22. Real Decreto 216/2014, de 28 de marzo, por el que se establece la metodología de cálculo de los precios voluntarios para el pequeño consumidor de energía eléctrica y su régimen jurídico de contratación. 2014.

23. **Ministerio de Industria, Turismo y Comercio.** Real Decreto 647/2011, de 9 de mayo, por el que se regula la actividad de gestor de cargas del sistema para la realización de servicios de recarga energética. 2011. 122.

24. —. Orden ITC/2794/2007, de 27 septiembre, por la que se revisan las tarifas eléctricas a partir del 1 de octubre de 2007. 2007. 234.

25. **Petrevski, Uros.** GitHub. [En línea] 8 de Mayo de 2013. <https://github.com/nodesign/MCP3008>.

26. **Syahidillah, Wildan Maulana.** GitHub. [En línea] 5 de Junio de 2017. <https://github.com/wildan2711/mqtt-android-tutorial>.

27. RS Components, resistencia 301k. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/resistencias-fijas-de-orificio-pasante/4778599/>.

28. RS Components, Resistencia 1,3k. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/resistencias-fijas-de-orificio-pasante/0148534/>.

29. RS Components, Resistencia 4,32k. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/resistencias-fijas-de-orificio-pasante/4778123/>.

30. RS Components, Resistencia 1k. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/resistencias-fijas-de-orificio-pasante/0157446/>.

31. RS Components, Condensador 220microF. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/condensadores-de-aluminio/1733115/>.

32. Amazon, esp8266. [En línea] [Citado el: 24 de 06 de 2018.] [https://www.amazon.es/AZDelivery-NodeMCU-ESP8266-ESP-12E-Desarrollo/dp/B06Y1ZPNMS/ref=sr\\_1\\_1\\_sspa?ie=UTF8&qid=1529838987&sr=8-1-spons&keywords=esp8266&psc=1](https://www.amazon.es/AZDelivery-NodeMCU-ESP8266-ESP-12E-Desarrollo/dp/B06Y1ZPNMS/ref=sr_1_1_sspa?ie=UTF8&qid=1529838987&sr=8-1-spons&keywords=esp8266&psc=1).

33. RS Components, MCP 3304. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/convertidores-analogico-a-digital-para-uso-general/0402762/>.

34. RS Components, Cable Unipolar. [En línea] [Citado el: 24 de 06 de 2018.] <https://es.rs-online.com/web/p/cables-unipolares-de-control/7244305/>.



## Anexos



ENGLISH

# Datasheet

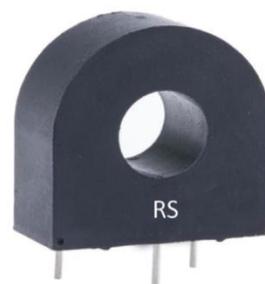
Stock No: 1243889

# RS Pro

## 50/60Hz Current Transformers

### Application

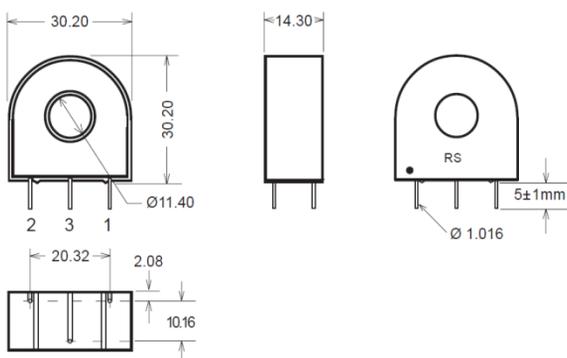
- Sensing Overload Current
- Ground fault detection
- Metering
- Analog to Digital Circuits
- Fully ROHS & REACH Compliance
- All Materials UL94-V0
- Storage temperature -55°C to +130°C



### Electrical Specifications @ 20°C ambient

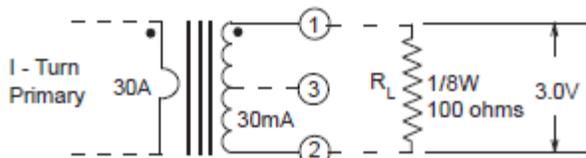
Primary Current	30A nom., 75A max.
Turns Ratio	1000:1 nominal
Volt per Amp Ratio at 30A for 100 Ω load	0.100 V/A
Volt per Amp Ratio at 3A for 100 Ω load	0.098 V/A
DC Resistance at 20°C	48 Ω
Dielectric Withstanding Voltage (Hi-pot)	4KVrms

### Mechanical Dimensions



### Notes:

- 1) Unless requested, the terminating resistor and the one-turn primary are not supplied
- 2) Pin 3: Normally for mechanical support only but will be used on center tapped designs
- 3) Tolerance  $\pm 0.20\text{mm}$

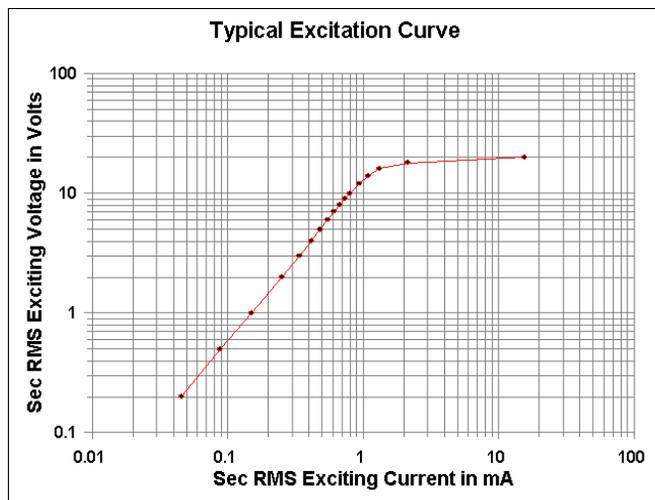
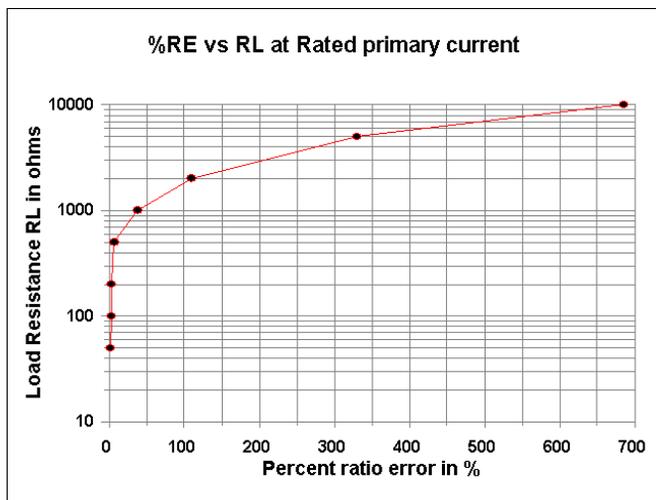
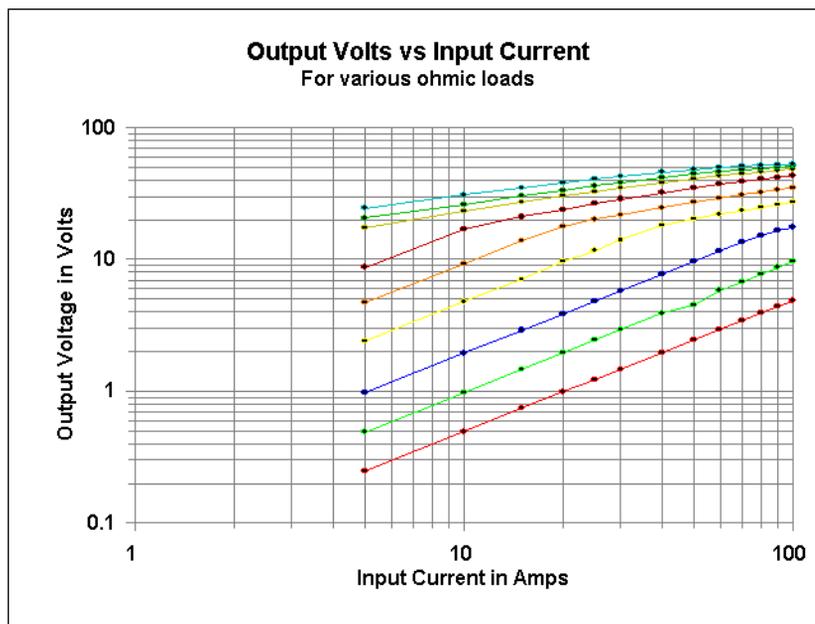




# Response Curves

AC1030

ENGLISH





### 3.0 PIN DESCRIPTIONS

The descriptions of the pins are listed in Table 3-1.

TABLE 3-1: PIN FUNCTION TABLE

MCP3302	MCP3304	Symbol	Description
PDIP, SOIC, TSSOP	PDIP, SOIC		
1	1	CH0	Analog Input
2	2	CH1	Analog Input
3	3	CH2	Analog Input
4	4	CH3	Analog Input
—	5	CH4	Analog Input
—	6	CH5	Analog Input
—	7	CH6	Analog Input
—	8	CH7	Analog Input
7	9	DGND	Digital Ground
8	10	CS/SHDN	Chip Select / Shutdown Input
9	11	D <sub>IN</sub>	Serial Data In
10	12	D <sub>OUT</sub>	Serial Data Out
11	13	CLK	Serial Clock
12	14	AGND	Analog Ground
13	15	V <sub>REF</sub>	Reference Voltage Input
14	16	V <sub>DD</sub>	+2.7V to 5.5V Power Supply
5, 6	—	NC	No Connection

#### 3.1 Analog Inputs (CH0-CH7)

Analog input channels. These pins have an absolute voltage range of  $V_{SS} - 0.3V$  to  $V_{DD} + 0.3V$ . The full scale differential input range is defined as the absolute value of  $(IN+) - (IN-)$ . This difference can not exceed the value of  $V_{REF} - 1$  LSB or digital code saturation will occur.

#### 3.2 Digital Ground (DGND)

Ground connection to internal digital circuitry. To ensure accuracy this pin must be connected to the same ground as AGND. If an analog ground plane is available, it is recommended that this device be tied to the analog ground plane in the circuit. See Section 5.6 "Layout Considerations" for more information regarding circuit layout.

#### 3.3 Chip Select/Shutdown ( $\overline{CS}/SHDN$ )

The  $\overline{CS}/SHDN$  pin is used to initiate communication with the device when pulled low. This pin will end a conversion and put the device in low-power standby when pulled high. The  $\overline{CS}/SHDN$  pin must be pulled high between conversions and cannot be tied low for multiple conversions. See Figure 6-2 for serial communication protocol.

#### 3.4 Serial Data Input (D<sub>IN</sub>)

The SPI port serial data input pin is used to clock in input channel configuration data. Data is latched on the rising edge of the clock. See Figure 6-2 for serial communication protocol.

#### 3.5 Serial Data Output (D<sub>OUT</sub>)

The SPI serial data output pin is used to shift out the results of the A/D conversion. Data will always change on the falling edge of each clock as the conversion takes place. See Figure 6-2 for serial communication protocol.

#### 3.6 Serial Clock (CLK)

The SPI clock pin is used to initiate a conversion and to clock out each bit of the conversion as it takes place. See Section 5.2 "Driving the Analog Input" for constraints on clock speed. See Figure 6-2 for serial communication protocol.

# MCP3302/04

## 6.2 Communicating with the MCP3302 and MCP3304

Communication with the MCP3302/04 devices is done using a standard SPI-compatible serial interface. Initiating communication with either device is done by bringing the  $\overline{CS}$  line low (see Figure 6-2). If the device was powered up with the  $\overline{CS}$  pin low, it must be brought high and back low to initiate communication. The first clock received with  $\overline{CS}$  low and  $D_{IN}$  high will constitute a start bit. The SGL/DIFF bit follows the start bit and will determine if the conversion will be done using single ended or differential input mode. Each channel in single ended mode will operate as a 12-bit converter with a unipolar output. No negative codes will be output in single ended mode. The next three bits (D0, D1, and D2) are used to select the input channel configuration. Table 6-1 and Table 6-2 show the configuration bits for the MCP3302 and MCP3304, respectively. The device will begin to sample the analog input on the fourth rising edge of the clock after the start bit has been received. The sample period will end on the falling edge of the fifth clock following the start bit.

After the D0 bit is input, one more clock is required to complete the sample and hold period ( $D_{IN}$  is a "don't care" for this clock). On the falling edge of the next clock, the device will output a low null bit. The next 13 clocks will output the result of the conversion with the sign bit first, followed by the 12 remaining data bits, as shown in Figure 6-2. Note that if the device is operating in the single ended mode, the sign bit will always be transmitted as a '0'. Data is always output from the device on the falling edge of the clock. If all 13 data bits have been transmitted, and the device continues to receive clocks while the  $\overline{CS}$  is held low, the device will output the conversion result, LSB, first, as shown in Figure 6-3. If more clocks are provided to the device while  $\overline{CS}$  is still low (after the LSB first data has been transmitted), the device will clock out zeros indefinitely.

If necessary, it is possible to bring  $\overline{CS}$  low and clock in leading zeros on the  $D_{IN}$  line before the start bit. This is often done when dealing with microcontroller-based SPI ports that must send 8 bits at a time. Refer to Section 6.3 "Using the MCP3302/04 with Microcontroller (MCU) SPI Ports" for more details on using the MCP3302/04 devices with hardware SPI ports.

TABLE 6-1: CONFIGURATION BITS FOR THE MCP3302

Control Bit Selections				Input Configuration	Channel Selection
Single /Diff	D2*	D1	D0		
1	X	0	0	single ended	CH0
1	X	0	1	single ended	CH1
1	X	1	0	single ended	CH2
1	X	1	1	single ended	CH3
0	X	0	0	differential	CH0 = IN+ CH1 = IN-
0	X	0	1	differential	CH0 = IN- CH1 = IN+
0	X	1	0	differential	CH2 = IN+ CH3 = IN-
0	X	1	1	differential	CH2 = IN- CH3 = IN+

\*D2 is don't care for MCP3302

TABLE 6-2: CONFIGURATION BITS FOR THE MCP3304

Control Bit Selections				Input Configuration	Channel Selection
Single /Diff	D2	D1	D0		
1	0	0	0	single ended	CH0
1	0	0	1	single ended	CH1
1	0	1	0	single ended	CH2
1	0	1	1	single ended	CH3
1	1	0	0	single ended	CH4
1	1	0	1	single ended	CH5
1	1	1	0	single ended	CH6
1	1	1	1	single ended	CH7
0	0	0	0	differential	CH0 = IN+ CH1 = IN-
0	0	0	1	differential	CH0 = IN- CH1 = IN+
0	0	1	0	differential	CH2 = IN+ CH3 = IN-
0	0	1	1	differential	CH2 = IN- CH3 = IN+
0	1	0	0	differential	CH4 = IN+ CH5 = IN-
0	1	0	1	differential	CH4 = IN- CH5 = IN+
0	1	1	0	differential	CH6 = IN+ CH7 = IN-
0	1	1	1	differential	CH6 = IN- CH7 = IN+

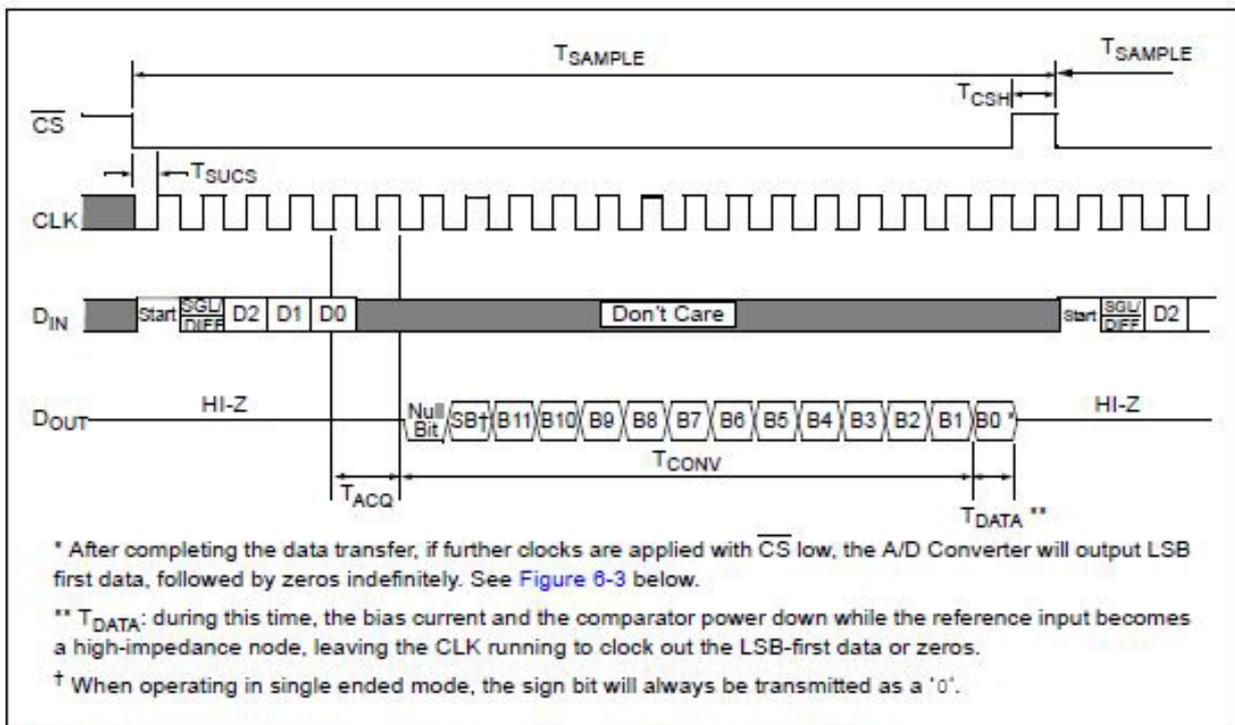


FIGURE 6-2: Communication with MCP3302/04 (MSB first Format).

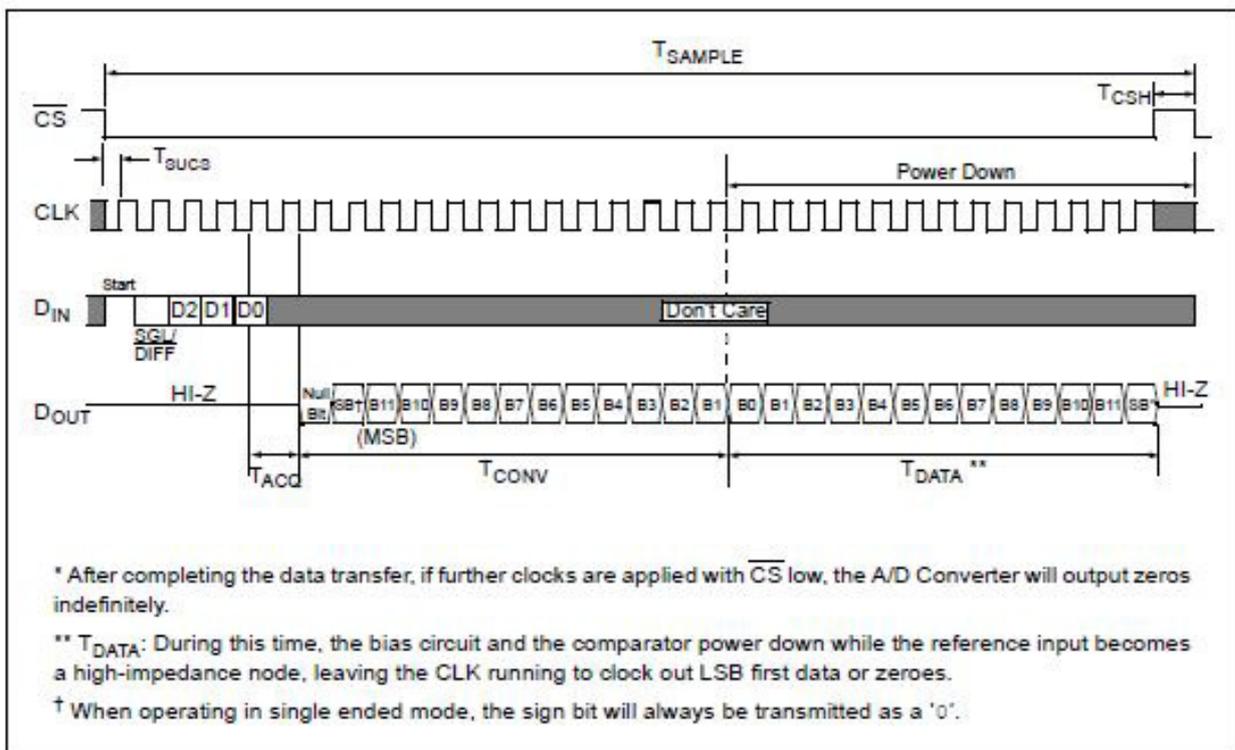


FIGURE 6-3: Communication with MCP3302/04 (LSB first Format).