

Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Monitor de corriente basado en Arduino™

Autor: Alberto Hermosín Gómez

Tutor: Ignacio Alvarado Aldea

Dpto. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2017





Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Monitor de corriente basado en Arduino™

Autor:

Alberto Hermosín Gómez

Tutor:

Ignacio Alvarado Aldea

Profesor contratado doctor

Dpto. de Ingeniería Automática y Sistemas

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Proyecto Fin de Carrera: Monitor de corriente basado en Arduino™

Autor: Alberto Hermosín Gómez

Tutor: Ignacio Alvarado Aldea

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal

A mi familia

A mis maestros

Resumen

El objeto de este Trabajo de Fin de Grado es el de proporcionar un aparato de medición y seguimiento del consumo de una máquina trifásica usada para hacer ensayos de tracción, perteneciente al dpto. de Ingeniería Mecánica, y solicitado expresamente a fin de conocer los distintos regímenes de consumo de dicha máquina. La forma adoptada para alcanzar este objetivo ha sido mediante una placa de desarrollo de la marca *Arduino*TM y el sensor de corriente inductivo SCT013-030.

Abstract

The aim of this End-of-Degree's Project is to provide a power consumption measurement and tracking device for a three-phase universal testing machine, located in the Mechanical Engineering laboratory, and which has been specifically requested to know in depth the power consumption of this machine under different working conditions. The resolution that has been taken is the utilization of an *Arduino*TM Launchpad and the inductive current sensor SCT013-030.

Índice

Resumen	viii
Abstract	x
Índice	xii
Índice de Tablas	xiv
Tabla de ilustraciones	xv
1 Introducción	1
1.1 <i>Motivación</i>	1
1.2 <i>Planteamiento del problema</i>	1
1.2.1 Información adicional	2
1.3 <i>Particularidades del problema</i>	2
1.3.1 Decisión final	3
2 Potencia activa y factor de potencia	4
3 Descripción del sistema de medida	7
3.1 <i>Estado del arte</i>	7
3.2 <i>Principales fabricantes</i>	7
3.3 <i>Fabricante elegido</i>	7
3.3.1 Motivos de la decisión	7
3.4 <i>Requisitos del sistema</i>	8
3.5 <i>Descripción por bloques</i>	9
3.6 <i>Descripción hardware</i>	10
3.6.1 El sensor de corriente	10
3.6.2 El circuito de acondicionamiento de señal	12
3.6.3 La placa <i>Arduino™</i> UNO	14
3.6.4 Pantalla LCD	16
3.6.5 Módulo de reloj RTC	17
3.6.6 Módulo MicroSD	18
3.6.7 Potenciómetro, pulsadores y diodos LED	18
3.6.8 El circuito de acondicionamiento de señal (versión inicial)	22
3.7 <i>Descripción software</i>	23
3.7.1 Inicialización de las comunicaciones, periféricos y módulos	25
3.7.2 Lectura del intervalo de captura	25
3.7.3 Comienzo y detención de ensayo	26
3.7.4 Comprobación de funcionamiento del módulo de reloj	26
3.7.5 El cálculo de la potencia y la energía	26
3.7.6 La lectura de fecha y hora	28
3.7.7 La escritura en el archivo	28
3.7.8 Librerías usadas	29
3.8 <i>Diseño de PCB</i>	29
3.9 <i>Diseño de carcasa</i>	31

4	Instalación y puesta en marcha	32
4.1	<i>Primer ensayo</i>	33
4.2	<i>Segundo ensayo</i>	34
5	Conclusiones y posibles ampliaciones	35
5.1	<i>Sistema remoto</i>	35
5.2	<i>Generación de archivos</i>	35
	Glosario	37
	Bibliografía	38

Índice de Tablas

Tabla 1. Desfases entre corriente (azul) y tensión (morado)

5

Tabla de ilustraciones

Ilustración 1.1. Máquina de ensayos de fatiga RUMUL TESTRONIC	1
Ilustración 1.2. Diagrama S-N (tensión-ciclos) de un ensayo de fatiga	2
Ilustración 2.1. (A la izquierda) Sistema de presurización.	4
Ilustración 2.2. (Arriba) Placa de características del subsistema principal.	4
Ilustración 2.3. (Abajo) Placa de características del subsistema de presurización.	4
Ilustración 2.4. Triángulo de potencias	5
Ilustración 2.5. Extracto de la hoja de características del sistema de presurización	6
Ilustración 3.1. Placa de pruebas <i>Arduino</i> TM UNO SMD edition	8
Ilustración 3.2. Esquema de bloques del sistema	9
Ilustración 3.3. Esquema del hardware del sistema	10
Ilustración 3.4. Sensor SCT013-030	10
Ilustración 3.5. PCB interna del SCT013-030 junto a secundario del transformador	11
Ilustración 3.6. Esquema equivalente	11
Ilustración 3.7. Equivalente completo	11
Ilustración 3.8. Adaptador para sensor	12
Ilustración 3.9. Montaje de sensor con adaptador	12
Ilustración 3.10. Circuito de acondicionamiento de la señal.	12
Ilustración 3.11. Salida de la etapa rectificadora. No hay caída de tensión respecto a la entrada.	13
Ilustración 3.12. Se observa la linealidad del circuito hasta que satura en 3.1 V	14
Ilustración 3.13. Esquema del ADC del ATmega328P	15
Ilustración 3.14. Display LCD	16
Ilustración 3.15. Interfaz I ² C-Paralelo. El potenciómetro se usa para ajustar el contraste de la pantalla.	17
Ilustración 3.16. Módulo RTC	17
Ilustración 3.17. Módulo MicroSD. Vista superior.	18
Ilustración 3.18. Módulo MicroSD. Vista inferior.	18
Ilustración 3.19. Esquemático final	20
Ilustración 3.20. Interconexión de la placa <i>Arduino</i> TM con todos los componentes.	21
Ilustración 3.21. Esquemático del circuito de acondicionamiento de señal	22
Ilustración 3.22. Señal del sensor antes de ser rectificada	22
Ilustración 3.23. Señal del sensor después de ser rectificada	22
Ilustración 3.24. Señal del sensor amplificada	23

Ilustración 3.25. Primera parte del diagrama de flujo del programa	23
Ilustración 3.26. Segunda parte del diagrama de flujo del sistema	24
Ilustración 3.27. Emulación de una señal rectificadora leída por el ADC de <i>Arduino</i> TM	27
Ilustración 3.28. Esquema interno del módulo RTC	28
Ilustración 3.29. Vista superior de la placa con las dos caras superpuestas. En rojo la cara superior y en azul la inferior.	30
Ilustración 3.30. Fotolito de cara de componentes	30
Ilustración 3.31. Fotolito de cara de pistas (reflejado)	30
Ilustración 3.32. Vista general de la carcasa del sistema	31
Ilustración 3.33. Vista inferior de la carcasa del sistema	31
Ilustración 3.34. Detalle de la perforación para alimentación	31
Ilustración 4.1. Tabla de especificaciones de la máquina de ensayos de fatiga	32
Ilustración 4.2. Representación del consumo eléctrico y la energía consumidas durante un ensayo de cinco minutos más otros cinco minutos previos al comienzo.	33
Ilustración 4.3. Imagen de la estructura del fichero generado	33

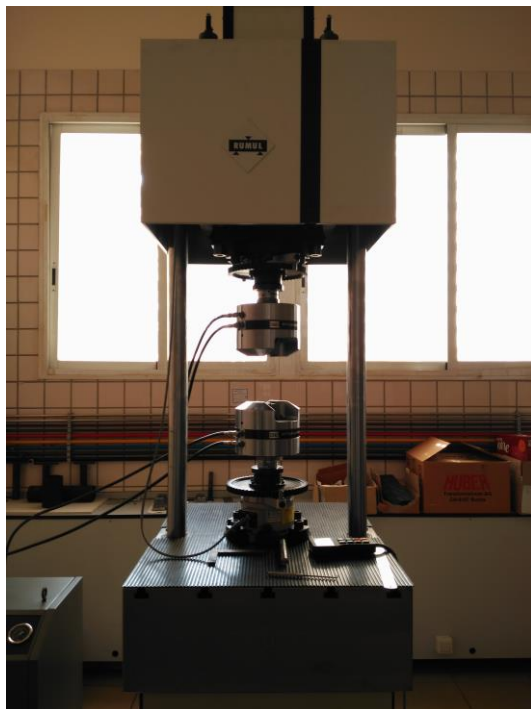
1 INTRODUCCIÓN

Este Trabajo de Fin de Grado, aunque bajo la tutela de un miembro del departamento de Ingeniería de Sistemas y Automática, se sitúa en el marco de la Ingeniería Electrónica Industrial de una forma esencialmente práctica. A lo largo de esta memoria se irá detallando el proceso del diseño conceptual, hardware, software y de acabado para un monitor de consumo basado en una placa *Arduino*TM UNO.

1.1 Motivación

La motivación de este trabajo es múltiple: en primer lugar, y como se detallará más adelante, el trabajo ha sido solicitado por necesidad de una persona, y cabe pensar que una de las principales funciones de la Ingeniería, en sentido amplio, es la de servir y ayudar a otras personas (entidades en general) a solucionar problemas concretos del día a día; en segundo lugar, esta necesidad se traducía en un trabajo en el que el peso de lo práctico era sustancialmente mayor que el de lo teórico. Esto, unido a una enseñanza universitaria con relativamente pocas sesiones de manejo de componentes electrónicos y diseño de circuitos, hizo que la realización de este proyecto despertase un especial interés en el autor. En tercer lugar, la posibilidad de desarrollar el trabajo casi en su totalidad de forma completamente autónoma (debido a que las circunstancias obligaron al autor a desarrollar el proyecto durante una estancia SICUE en Bilbao) fue un aliciente, así como poder comprobar personalmente la eficiencia del método de aprendizaje basado en la investigación. Por último, la toma de contacto con un entorno como es el de *Arduino*TM que es tan versátil y con tantos recursos a su disposición que no se entiende cómo se puede prescindir de él en un Grado de Ingeniería enfocado al desarrollo de las tecnologías industriales.

1.2 Planteamiento del problema



En su afán por desarrollar los trabajos e investigaciones docentes de su departamento de una manera lo más sostenible posible, el profesor contratado doctor Víctor M. Chaves, del dpto. de Ingeniería Mecánica de la Universidad de Sevilla, pidió que se desarrollase un sistema que le proporcionase información acerca del consumo eléctrico de una de las máquinas usadas en su trabajo.

Se trata de la máquina de la imagen a la izquierda, una máquina de ensayos de fatiga. Con ella se someten distintos materiales a esfuerzos cíclicos en direcciones opuestas (compresión y tracción) durante un número de repeticiones (ciclos).

Ajustando un **juego de pesas** en la caja superior se ajusta la frecuencia de los ciclos para la prueba, que puede variar entre los 40 y los 260 Hz.

Ilustración 1.1. Máquina de ensayos de fatiga
RUMUL TESTRONIC

1.2.1 Información adicional

En estas máquinas existen dos plataformas: una primera plataforma fija (la inferior) con una pinza para sostener un extremo del material a estudiar (en forma de pequeño cilindro estriado, llamado *probeta*) y una segunda plataforma móvil con otra pinza para sostener el otro extremo y que está programada para moverse a una frecuencia determinada por la colocación del juego de pesas de la parte superior. Los esfuerzos se generan cíclicamente en la dirección del eje de la probeta. De esta manera se obtiene una *curva tensión-ciclos*, que es un diagrama muy usado en el campo de la Ingeniería Mecánica a través del cual se pueden vislumbrar a simple vista muchas de las propiedades deseadas.

A lo largo de estos ensayos, que pueden durar desde varios minutos hasta varios días, la máquina va pasando por diferentes regímenes de funcionamiento y de consumo, en función de la resistencia que presente el material sometido a estudio. Más aún, ante estudios sobre el mismo material se deseaba saber si, entre otras cosas, la temperatura ambiente en una ciudad como Sevilla, que oscila entre los 10 °C de media en invierno y los 30 °C en verano, tiene mucha o poca influencia en los datos de consumo.

Como resultado final al hacer diferentes ensayos sobre el mismo material se obtiene una curva característica del material como la que se observa a continuación.

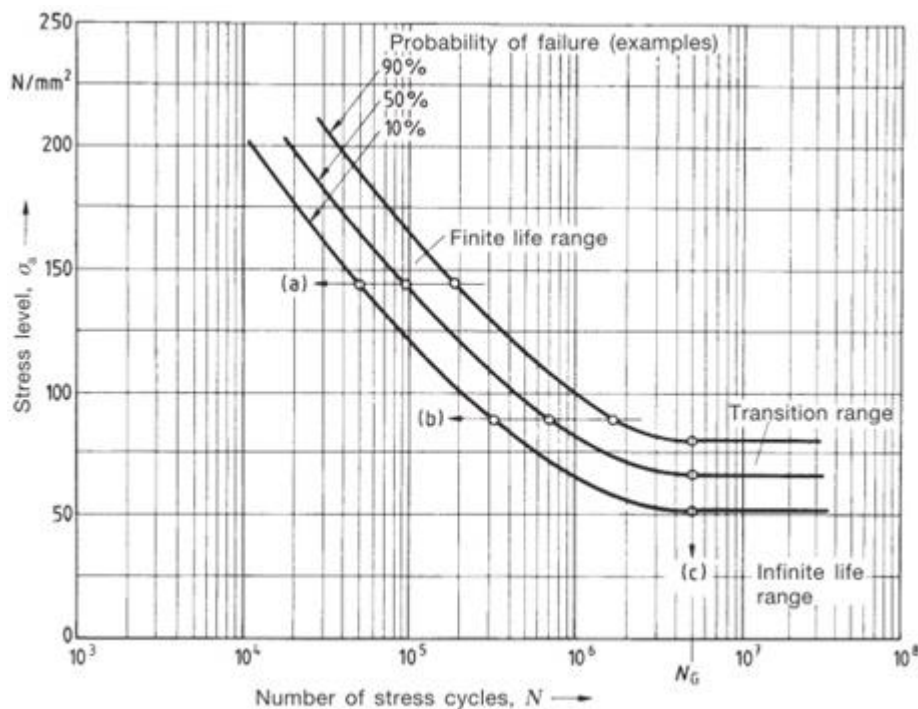


Ilustración 1.2. Diagrama S-N (tensión-ciclos) de un ensayo de fatiga

1.3 Particularidades del problema

Según el principio por el que se mueva la plataforma móvil de la máquina existen dos tipos:

- Máquinas hidráulicas: son las más voluminosas. En ellas los mecanismos de movimiento y de aplicación de fuerzas están accionados por fluidos que se mueven dentro de conductos. Estos fluidos vienen presurizados desde unas bombas de presión situadas, por lo general, en una sala distinta a la que alberga a la máquina.
- Máquinas eléctricas: accionada de manera enteramente eléctrica hasta la plataforma móvil.

En el caso de las máquinas hidráulicas se presenta una problemática añadida difícil de solventar a la hora de calcular su consumo. Esto ocurre porque el paso de la parte eléctrica a la parte hidráulica de estos sistemas no se hace de forma directa, ya que el cableado eléctrico se introduce en una caja negra inaccesible por los usuarios,

donde se encuentra el sistema de bombas y de refrigeración, además de que el mismo sistema de bombas de presión alimenta no sólo a una máquina sino a varias a la vez. Por otro lado, lo deseable sería poder medir cada uno de los componentes dentro de esa caja para saber la influencia de cada uno: bombas, sistemas de refrigeración y cualquier otro tipo de aparataje eléctrica que pudiera haber en su interior.

1.3.1 Decisión final

Vistas estas dificultades, se decidió que en lugar aplicar el sistema de medición de consumo a las máquinas hidráulicas se empezara por hacerlo con las eléctricas, ya que su conexión a la red tiene la posibilidad de ser en monofásico o en trifásico, y para los ensayos que se realizan en el departamento se usa el modo monofásico.

2 POTENCIA ACTIVA Y FACTOR DE POTENCIA

Para medir la potencia activa en un sistema que está accionado por un motor de corriente alterna se precisan saber varios datos, que se pueden adquirir ya sea mediante un libro de especificaciones del fabricante o leyendo la placa de características de dicho motor. En este caso ha habido que recurrir a las dos opciones.

El sistema a medir se compone a su vez de dos subsistemas:

- El subsistema principal es el que contiene la base y las pinzas para sujetar el material del ensayo, como se ve en la **Ilustración 1.1**.
- El subsistema secundario está formado por una caja que alberga un depósito de un fluido y un motor para presurizarlo, mostrado en la **Ilustración 2.1**; **Error! No se encuentra el origen de la referencia.**

En el subsistema principal hay dos agentes de consumo integrados en la misma caja: por un lado, el motor eléctrico de corriente alterna que acciona el sistema de pesas situado en la parte superior de la máquina. Por otro lado, hay un sistema de ventilación integrado que está en continuo funcionamiento.

En el subsistema secundario sólo actúa el motor que presuriza el fluido del depósito. Este fluido presurizado se usa para accionar las pinzas que sostienen la probeta sometida a ensayo de una manera firme. Para ello, el motor entra en un régimen de funcionamiento absolutamente constante determinado por las características del fabricante en su libro de especificaciones como de **750 W**. Aun así, el sistema presenta una placa de características en la que se proporcionan valores de tensión e intensidad consumidas.

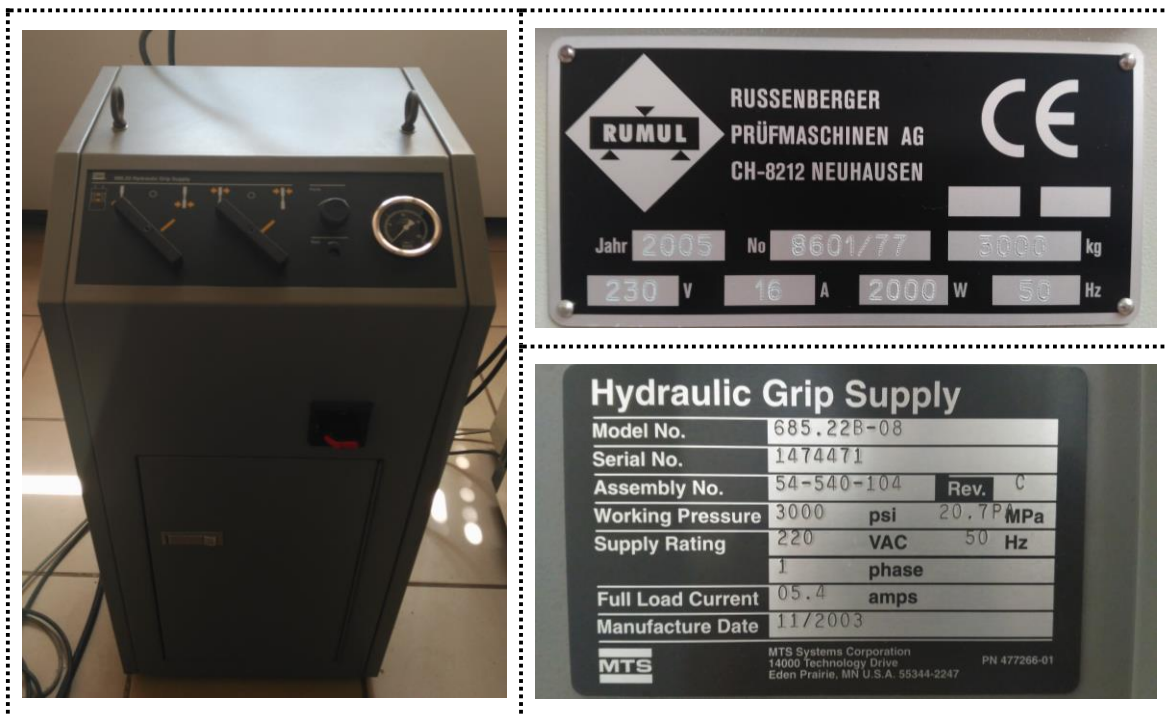


Ilustración 2.1. (A la izquierda) Sistema de presurización.

Ilustración 2.2. (Arriba) Placa de características del subsistema principal.

Ilustración 2.3. (Abajo) Placa de características del subsistema de presurización.

El factor de potencia en un sistema eléctrico alterno es una consecuencia natural de la existencia de componentes inductivos y capacitivos. Según las ecuaciones que modelan a las resistencias, los condensadores y las bobinas, la corriente respecto a la tensión puede estar en tres situaciones:

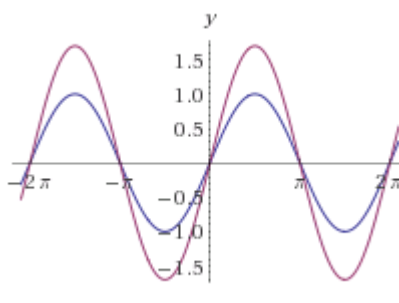
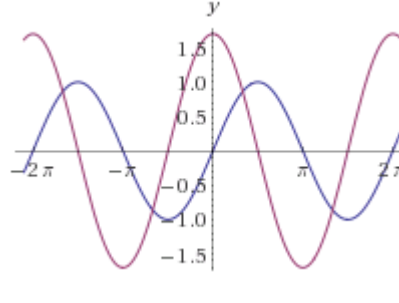
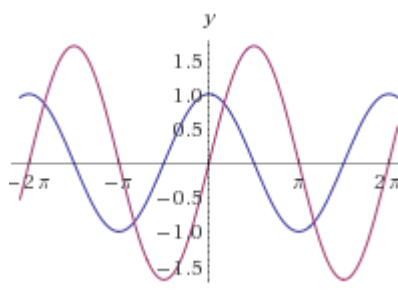
		
<p>En un circuito puramente resistivo la corriente y la tensión están en fase.</p>	<p>En los circuitos puramente capacitivos la corriente adelanta en $\pi/2 \text{ rad}$ a la tensión.</p>	<p>En los circuitos puramente inductivos la corriente se retrasa en $\pi/2 \text{ rad}$ a la tensión.</p>

Tabla 1. Desfases entre corriente (azul) y tensión (morado)

Los componentes inductivos (bobinas) y capacitivos (condensadores) no consumen potencia activa, sino reactiva. Por tanto, surge una expresión distinta de la potencia: la potencia aparente:

$$S = I * U$$

Donde

S es el fasor¹ de potencia aparente, y se mide en voltio amperio (VA)

I es el fasor de corriente eficaz (RMS)

U es el fasor de tensión eficaz (RMS)

Así, S es un fasor cuyas dos componentes son la potencia activa (P , medida en vatios, W) y la potencia reactiva (Q , medida en voltio amperio reactivo, VAR), según el siguiente triángulo de potencias:

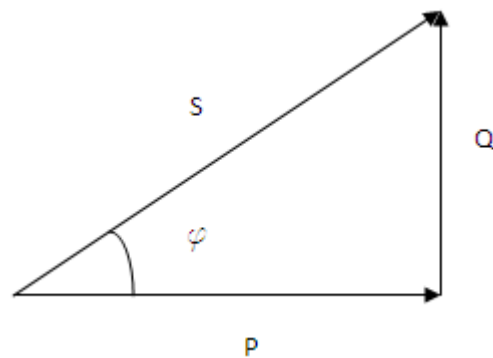


Ilustración 2.4. Triángulo de potencias

De esta manera se observa que la relación entre la potencia aparente de una máquina y su potencia activa, que es la que interesa medir, es el ángulo que existe entre ellas, ϕ , según la siguiente expresión:

$$P = |S| \cos(\varphi) = |I| |U| \cos(\varphi)$$

Donde **$\cos(\varphi)$** es el factor de potencia.

¹ Los fasores representan las ondas eléctricas en el tiempo como números complejos con los que es más fácil hacer los cálculos en los circuitos.

Debido a este hecho, la mayoría de los fabricantes de motores de corriente alterna, entre otros dispositivos, incluyen como información imprescindible el factor de potencia. Éste, junto a los valores de tensión y de corriente nominales, nos da la información necesaria para calcular el consumo de potencia activa de la máquina. Mientras que la tensión y corriente que alimentan al motor pueden sufrir ligeras variaciones, el factor de potencia es una característica inherente a éste.

Precisamente, en ninguno de los dos motores que albergan los dos subsistemas que se pretenden monitorizar se ha facilitado esa información directamente en sus placas características, así que habrá que hacer una estimación en función de los datos nominales. Así, se tiene que en el subsistema principal:

$$\cos(\varphi_1) = \frac{P}{UI} = \frac{2000 W}{220 V * 16 A} = 0.57$$

Con este valor calculado podemos tener una primera estimación del consumo que se verá reflejado en los ensayos.

Por otro lado, el subsistema de presurización tiene sólo una tensión nominal a la que estar conectado y una corriente de máxima carga. Con estos datos no se puede obtener el factor de potencia, ya que significaría que la máquina estaría trabajando siempre al máximo de su capacidad. Sin embargo, haciendo una búsqueda en Internet proporcionando el fabricante y el modelo de la máquina se puede encontrar una hoja de características en la que indica un consumo nominal, por lo que habrá que realizar ensayos y calibrar el sistema de medida en función de ellos. Esto se verá en los últimos apartados.

Model 685.10 and Model 685.22 Standalone Hydraulic Grip Supplies

The 685.10 and 685.22 units feature a self-contained hydraulic pump, a 0.75 kW (1 hp) electric motor, a 11.3 l (3 gal) reservoir, a 10-micron absolute return line filter, and hoses for connection to grips. These units

Ilustración 2.5. Extracto de la hoja de características del sistema de presurización

3 DESCRIPCIÓN DEL SISTEMA DE MEDIDA

3.1 Estado del arte

A lo largo de los últimos años se ha venido desarrollando toda una gama de productos electrónicos de diversos fabricantes y precios, pero basados todos en el mismo principio: el uso de un microcontrolador. El primer microcontrolador data de enero de 1971, fecha desde la cual han ido evolucionando de manera apreciable.

Hoy en día la tecnología que los gobierna ha cambiado considerablemente, así como el manejo de los mismos. De esta manera, tenemos distintos fabricantes con sus propios productos en forma de placas de pruebas, a modo de pequeños sistemas embebidos. Desde ellas, se puede programar el microcontrolador que actúa de cerebro para fines tanto a nivel de aficionado como a nivel profesional, dependiendo del producto en concreto.

3.2 Principales fabricantes

Así, contamos con algunos nombres conocidos de fabricantes que se dedican a hacer productos de características similares:

- *Texas Instruments Inc.*, con su placa de desarrollo basado en la familia de microcontroladores MSP430.
- *Raspberry Pi*, que se dedica íntegramente a la producción de SBC. Estos sistemas no sólo tienen sus microcontroladores y periféricos, sino que además cuentan con un pequeño sistema operativo integrado.
- *Arduino*TM, popularizado como vendedor para aficionados a la electrónica además de entornos más profesionales.

Cada uno de ellos utiliza diferentes sistemas a la hora de programarlos. Las placas de desarrollo de *Texas Instruments* tienen su propio IDE, el software *Code Composer Studio*, donde el código se puede depurar y a través del cual se transmite el programa a la memoria del microcontrolador. En *Raspberry Pi* se programa directamente a través del *shell*, una pantalla interactiva similar a la que se utiliza en MATLAB. Los productos de *Raspberry Pi* están más enfocados al manejo y procesamiento de grandes cantidades de datos, lo que hace su uso más frecuente en el ámbito de las telecomunicaciones y las comunicaciones industriales.

3.3 Fabricante elegido

El caso que nos ocupa es el último, en el que es quizá su producto más básico. *Arduino*TM UNO es el producto del fabricante *Arduino*TM más limitado en cuanto a funciones y pines accesibles.

3.3.1 Motivos de la decisión

El hecho de decidirse por este fabricante es que, mientras que como se dijo en el apartado anterior, los productos como *Raspberry Pi* están más enfocados al software (procesamiento de grandes flujos de datos) la gama de *Arduino*TM está más optimizada en tanto al uso de sensores y periféricos de diferente índole (hardware en general). Por otro lado, es más fácil encontrar componentes y módulos compatibles para *Arduino*TM más que para los productos de *Texas Instruments Inc.*, por lo que se tomó esta decisión.

La programación de las placas *Arduino*TM también se hace a través de un IDE propio, gratis, ligero y descargable desde su web, y que utiliza un sublenguaje de C++, de más alto nivel que el lenguaje C, con una extensa colección de librerías y funciones. Esta cantidad de material se ha conseguido principalmente por la filosofía a la que se adhiere la marca *Arduino*TM: el código abierto (Open Source). El principio fundamental del código abierto es que todo el código que forma parte de un producto en todas sus vertientes es accesible por cualquier usuario de forma gratuita, de forma que se pueden hacer modificaciones de cualquier parte del código. Así, miles de usuarios a lo largo de los últimos años han desarrollado librerías dedicadas a diferentes protocolos de

comunicaciones, a manejo de distintos periféricos, sensores, motores, etc.

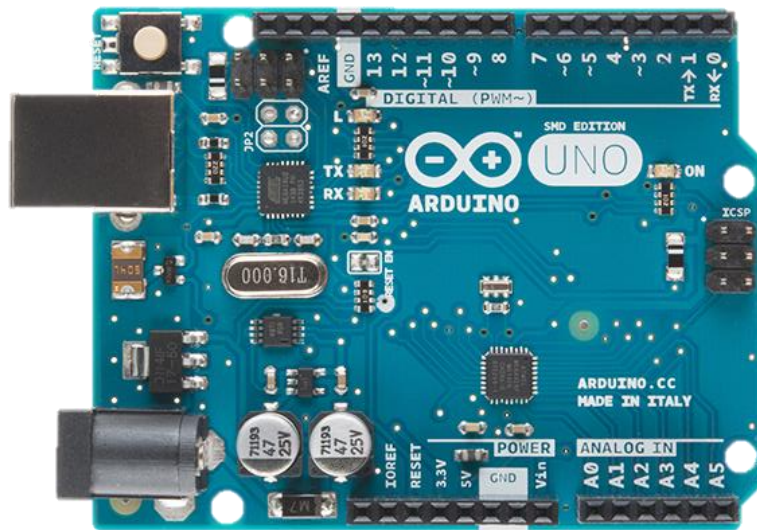


Ilustración 3.1. Placa de pruebas *Arduino*TM UNO SMD edition

3.4 Requisitos del sistema

Según los requerimientos del Prof. Víctor Chaves, se convino en que el sistema objeto del Trabajo de Fin de Grado habría de tener los siguientes requisitos:

- Medición en tiempo real de consumo eléctrico en kilovatios.
- Lectura en pantalla de LCD de dicho consumo.
- Capacidad de generar archivos de texto plano para posteriormente procesar la información del consumo medido con hora y fecha.
- Capacidad de decidir cuándo generar esos archivos de texto plano (en adelante, *ensayos*) mediante botones (“Comenzar ensayo” y “Terminar ensayo”).
- Posibilidad de programar el intervalo de captura de datos de consumo mediante un potenciómetro (1 s, 2 s, 5 s y 10 s).
- Volcado de datos hacia una tarjeta de memoria MicroSD.
- Procesado de archivos de texto plano mediante Microsoft Office Excel para mejor visualización de los datos obtenidos.
- Colocación de todo el sistema en una carcasa de fácil manejo y tamaño reducido.

3.5 Descripción por bloques

Para empezar, se hará un esquema de bloques en el que se ilustre cómo el sistema ha de ir cumpliendo con cada uno de los requisitos anteriores.

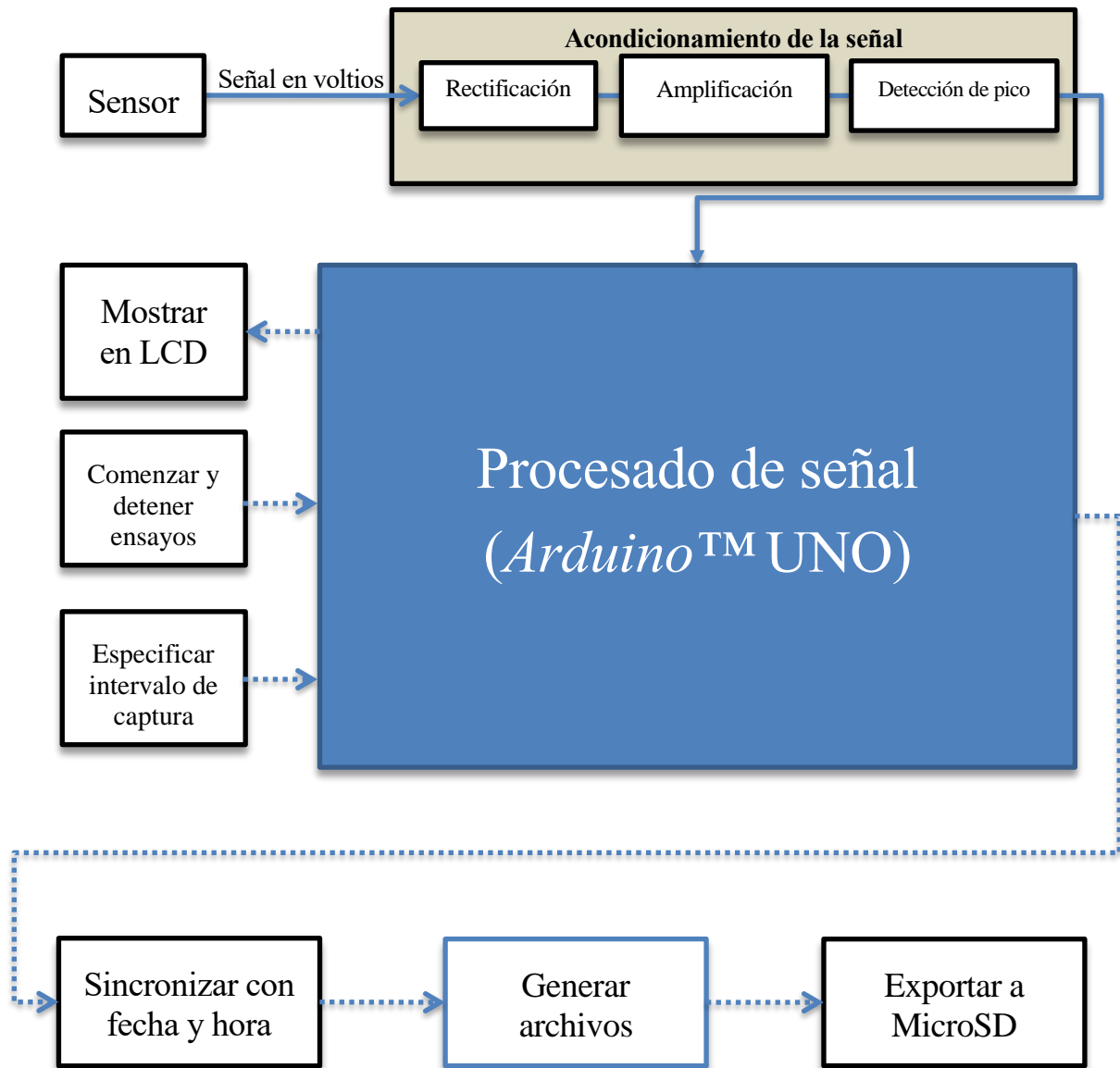


Ilustración 3.2. Esquema de bloques del sistema

Dado este esquema se ve cómo se subsanan todas las necesidades anteriormente descritas en los requisitos del sistema. En él, los cuadros con el marco en negro se refieren a la parte de hardware del proyecto, y los cuadros con el marco en azul a la parte de software. Por otro lado, las flechas continuas indican magnitud continua (tensión) y las flechas punteadas, magnitud cuantizada (bits). En los siguientes apartados se verá con qué componentes y de qué maneras se van consiguiendo estas funciones.

3.6 Descripción hardware

Una vez vista la descripción por bloques cabe hacer una segunda descripción que defina todo el hardware que conlleva lo anterior. En esta subsección también se detallarán los elementos básicos de todo el montaje, y más en profundidad los más importantes, como son el sensor de corriente y la propia placa de *Arduino*TM.

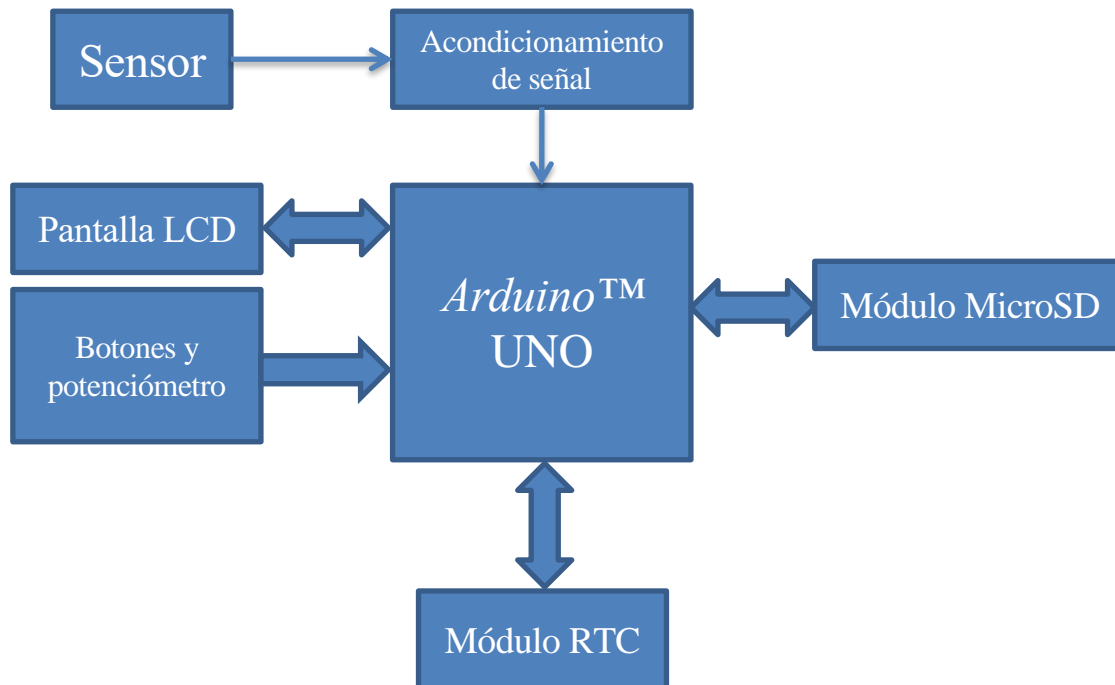


Ilustración 3.3. Esquema del hardware del sistema

A continuación, se pasará a detallar cada uno de estos bloques y componentes.

3.6.1 El sensor de corriente

El sensor que se ha escogido para hacer posible el cálculo de la potencia consumida es el SCT013-030, un transformador de corriente-tensión para corriente alterna con una relación nominal de 30 A: 1 V (es decir, que para una entrada de 30 A genera una salida de 1 V, según la hoja de características). El sensor en cuestión se muestra en la **Ilustración 3.4**:



Ilustración 3.4. Sensor SCT013-030

El modo de empleo para este tipo de sensores (de pinza amperimétrica) es el siguiente:

- Se abre la pinza por la pestaña indicada.
- Se toma **un solo hilo** de la alimentación a la que está conectada el dispositivo cuyo consumo se desea medir.
- Se hace pasar por el centro de la pinza.
- Se cierra la pinza.

La tensión de salida será medida según un cable de audio estándar de 3.5 mm.

Este sensor presenta en su interior un circuito mínimo en el que se puede apreciar cómo se hace la conversión de corriente a tensión. En primer lugar, los dos bobinados del transformador lo forman, por un lado, el propio cable por el que pasa la corriente que se va a medir (primario) y, por otro lado, un hilo de cobre de 1 mm con un total de 1800 vueltas (secundario) sobre un núcleo de ferrita que se puede abrir para dejar pasar el cable (como se detalló anteriormente). El circuito posterior, como se puede apreciar en la **Ilustración 3.5**, está construido sobre una placa de circuito impreso (PCB) multipropósito en la que muchas de las pistas quedan inutilizadas para esta funcionalidad, dejando un equivalente eléctrico tan sencillo como el que se observa en la **Ilustración 3.6**, extraída directamente de la hoja de características del fabricante del sensor.

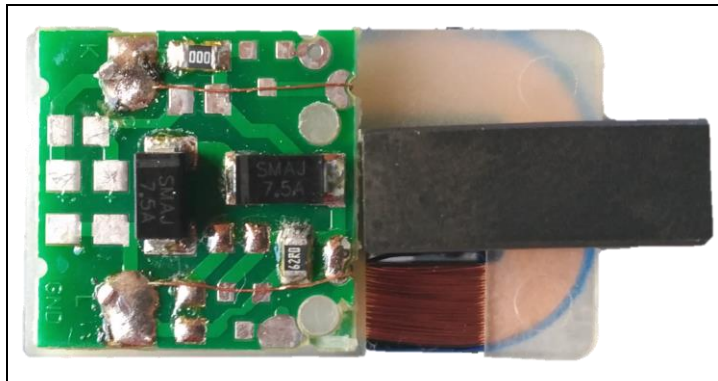


Ilustración 3.5. PCB interna del SCT013-030 junto a secundario del transformador

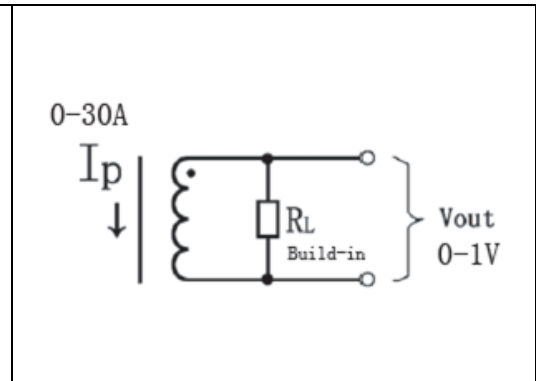


Ilustración 3.6. Esquema equivalente

Los componentes de mayor tamaño que se observan en la imagen anterior son dos diodos SMD que cumplen una función de protección frente a posibles sobreintensidades en el momento de estar midiendo. Se obvian en el esquema porque en funcionamiento normal del dispositivo no actúan de ninguna manera, pero están colocados en paralelo a la resistencia de salida R_L en disposición de cátodos enfrentados, limitando así la salida a un máximo de aproximadamente ± 7.5 V según el fabricante. Por otro lado, se observa una resistencia de 0Ω que une dos pistas, para el caso de esta PCB, con un simple cortocircuito. El esquema completo con estos componentes de más se aprecia en la figura de la izquierda.

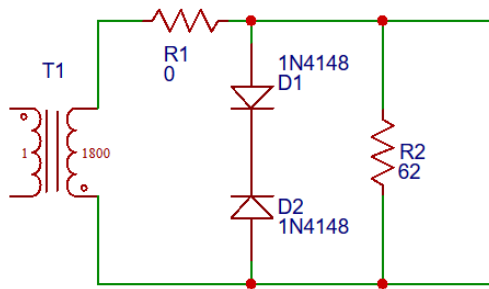


Ilustración 3.7. Esquema completo

El primario del transformador representa el hilo por el que pasa la corriente que genera el consumo y el secundario es el bobinado de 1800 vueltas.

Se puede hacer un desarrollo sencillo para ilustrar cómo los valores proporcionados de la transformación implican que haya que usar los valores que se muestran en la ilustración.

De la relación de transformación se tiene la expresión siguiente:

$$\frac{N_1}{N_2} = \frac{U_1}{U_2} = \frac{I_2}{I_1}$$

Por lo tanto, $I_2 = I_1 \frac{N_1}{N_2}$, con $N_1 = 1$ y $N_2 = 1800$.

De esta manera, para $I_1 = 30$ A $\rightarrow I_2 = \frac{1}{1800} 30$ A = 0.0167 A, y así:

$$V_{out} = I_2 R_2 = 0.0167 * 62 \approx 1$$
 V

3.6.1.1 Adaptador para el sensor SCT013-030

Como se dijo anteriormente, el sensor necesita que sólo un hilo del cable que alimente a una carga (de forma genérica) pase por su interior. Por lo general, los cables de alimentación tienen dos o más cables en el caso de cargas monofásicas: ida, retorno y, a veces, tierra de protección. Para hacer esto habría que quitar la cubierta que tiene el cable que envuelve a los dos (o tres) conductores, y posteriormente rodear con la pinza la ida o el retorno

(según la primera ley de Kirchhoff, la intensidad por un hilo de ida es la misma que la de vuelta, ya que la tierra de protección en condiciones normales arrastra una corriente nula).

Para subsanar este hecho se fabrica un pequeño adaptador que sirve para la fase de pruebas con cargas monofásicas que tiene el siguiente aspecto:



Ilustración 3.8. Adaptador para sensor



Ilustración 3.9. Montaje de sensor con adaptador

Como se puede observar, se trata simplemente de una extensión del enchufe al que se va a conectar. En un momento del cable (el final, por simplicidad en el montaje) se quita el recubrimiento blanco que agrupa a los conductores de ida, vuelta y protección. Sabiendo que el de protección es, por código de colores, el amarillo-verde, basta con colocar el sensor alrededor de cualquiera de los otros dos conductores (en el caso de las imágenes, el azul). De esta manera, se estará midiendo la intensidad que le está llegando a la carga conectada a esa toma de corriente.

3.6.2 El circuito de acondicionamiento de señal

Es la parte de electrónica más analógica del proyecto. Dado que la placa de *Arduino*TM va a leer una señal analógica mediante el convertidor Analógico-Digital (ADC) que tiene incorporado, es conveniente que esta señal sea lo más estable y constante posible, por lo que el acondicionamiento de la tensión de salida del sensor pasa por tres etapas, resultando un esquema como el siguiente:

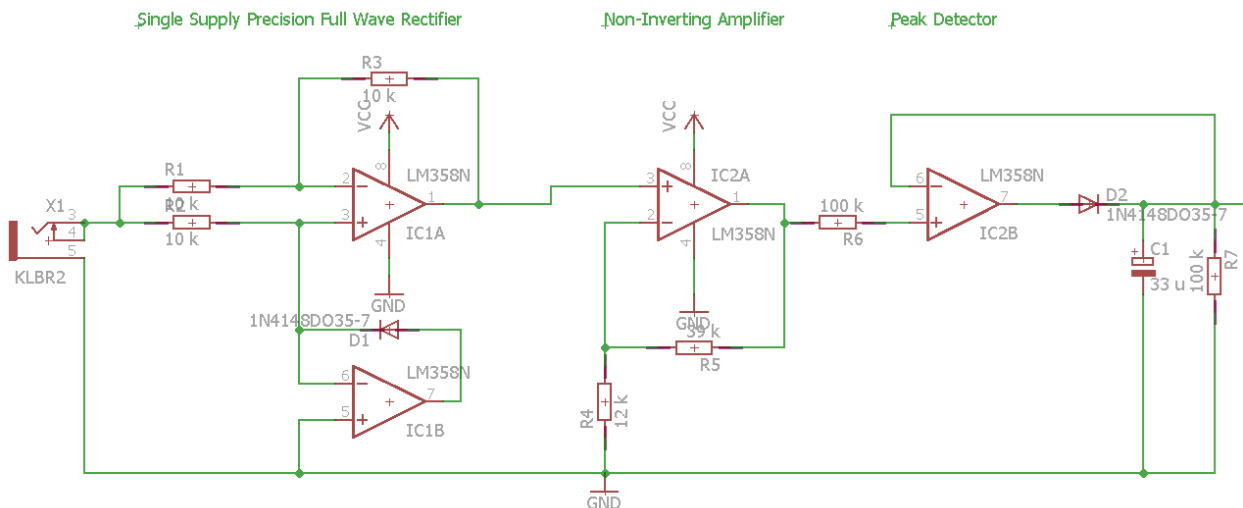


Ilustración 3.10. Circuito de acondicionamiento de la señal.

En primer lugar, un rectificador de precisión de onda completa y alimentación simple se encarga de obtener una señal con el valor absoluto de la señal sinusoidal. Este rectificador está constituido por dos amplificadores operacionales, tres resistencias de igual valor y un diodo. En el semiperiodo positivo de la onda de entrada, el amplificador operacional de abajo (IC1B) tiene una tensión mayor en el pin negativo que en el pin positivo, forzando su salida a la alimentación inferior (tierra, GND). Así, el diodo queda polarizado inversamente y no deja fluir corriente a través de él. Considerando a su vez que la impedancia en las entradas de un amplificador operacional es prácticamente infinita, durante el semiperiodo positivo esta parte del circuito (amplificador y diodo) es como si no estuviera, quedando solamente la parte superior, formada por las resistencias R_1 , R_2 y R_3 . Si se observa la configuración que tiene se deduce que se corresponde con la de un amplificador no inversor, y

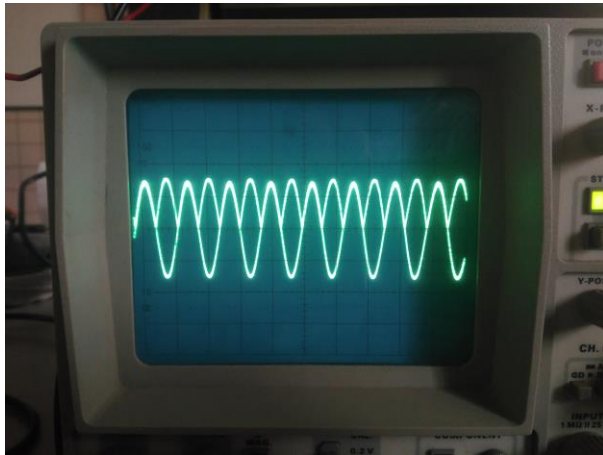


Ilustración 3.11. Salida de la etapa rectificadora. No hay caída de tensión respecto a la entrada.

por tener todas las resistencias iguales, de ganancia unidad. En el semiperiodo negativo de la señal, el diodo del lazo de realimentación entra en polarización directa y hace que el pin de la entrada no inversor del amplificador IC1A esté conectado virtualmente a tierra. De esta manera, la señal pasa directamente a la entrada inversora del amplificador, y de manera análoga al semiperiodo anterior se consigue una configuración inversora de ganancia unidad. En resumen, la parte inferior del circuito sirve para conmutar a la parte superior entre amplificador de ganancia unidad inversor y no inversor.

En segundo lugar, la señal pasa por un amplificador no inversor de ganancia fija $\left(1 + \frac{R_5}{R_4}\right)$, con $R_4 = 12\text{ k}\Omega$ y $R_5 = 39\text{ k}\Omega$, con lo que se consigue una ganancia teórica de 4.25. En la práctica se genera un pequeño error debido a las no idealidades de los componentes, dejando la ganancia en **4.17**. Esta ganancia no es trivial, ya que se ha obtenido mediante ensayos. El hecho de escoger esa ganancia está condicionado por dos factores: por un lado, la salida máxima que puede dar el sensor (algo menos de 1 V), y por otro lado la salida máxima que puede dar el circuito de acondicionamiento, que junto con la fase siguiente tiene un límite de saturación en 3.1 V aproximadamente. En este valor de la ganancia se garantiza que se optimiza el rango útil de entrada y el rango posible de salida del circuito de acondicionamiento.

En la última etapa, la señal pasa por un detector de pico. El funcionamiento de un circuito de detección de pico es sencillo de entender. El circuito funciona como un seguidor de tensión en un primer momento, por lo tanto, a la salida se introduce una tensión que va desde 0 V hasta un valor de pico, ya que tenemos una señal rectificadora a onda completa. Durante este periodo de tiempo, esa tensión positiva a la salida hace que, junto al diodo polarizado directamente y la resistencia que sirve de camino de salida de la corriente, el condensador se vaya cargando hasta un máximo. Cuando la tensión a la entrada empieza a descender, la salida tratará de seguirla al igual que antes, pero debido a la carga almacenada en el condensador la tensión decrece más lentamente, exponencialmente según una expresión del tipo $v_c = v_i e^{-\frac{t}{\tau}}$. En este momento, el diodo se polariza inversamente, por lo que el condensador debe descargar sobre la resistencia R_7 que tiene situada en paralelo y la salida.

Así, en la salida de este circuito se obtiene una señal con un rizado imperceptible para las medidas que necesitamos. El coste de esta rectificación casi absoluta es una dinámica lenta de la respuesta para cambios muy bruscos en el consumo y sean descendentes hacia el cero (consumo nulo).

Realizando pruebas con una fuente de tensión a modo de salida del sensor se obtiene una gráfica como la siguiente:

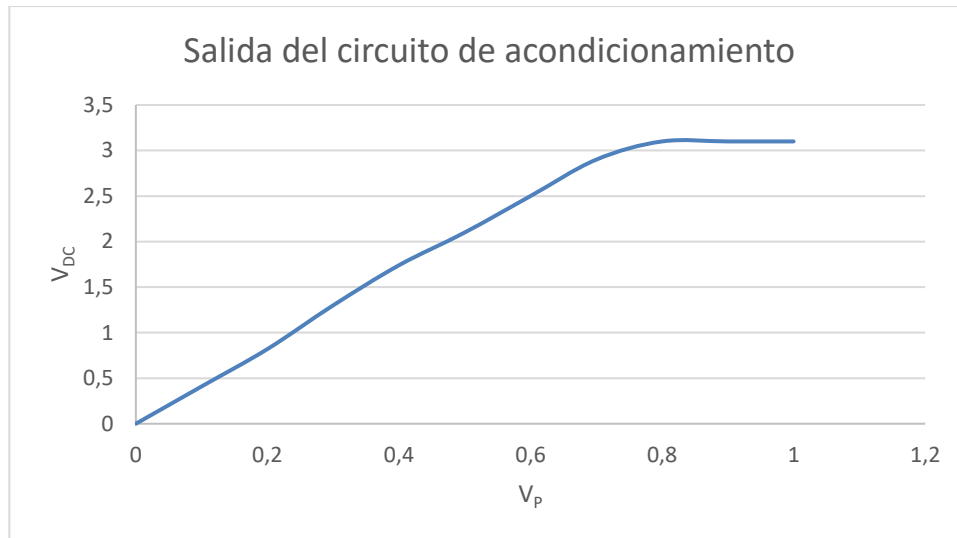


Ilustración 3.12. Se observa la linealidad del circuito hasta que satura en 3.1 V

3.6.3 La placa *Arduino*™ UNO

En este apartado se resaltarán algunos de los aspectos más relevantes de la placa de pruebas *Arduino*™ UNO con respecto a este proyecto, así como las conexiones entre dicha placa y los módulos y componentes externos utilizados.

Arduino™ UNO es una placa basada en el microcontrolador ATmega328P. Éste posee un oscilador de cuarzo de 16 MHz, 14 entradas/salidas digitales (con 6 que se pueden usar como PWM para emular una salida analógica, aunque en este proyecto no se usan), 6 entradas analógicas (con lo que podremos proporcionar una entrada a nuestra señal rectificadora).

Estas entradas analógicas son procesadas por un convertidor analógico-digital que trae la placa incorporado, que presenta el siguiente esquema:

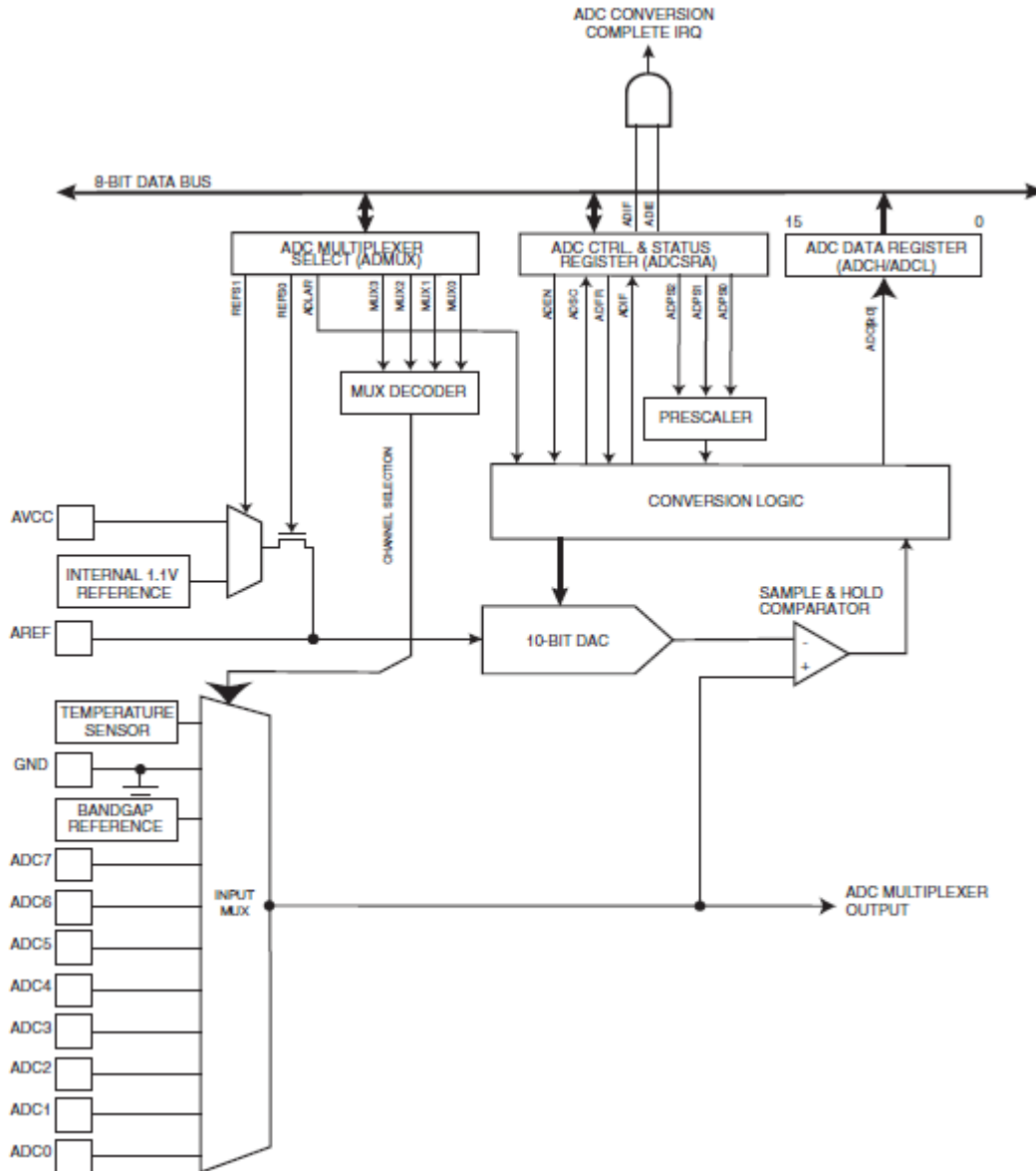


Ilustración 3.13. Esquema del ADC del ATmega328P

Los datos que hay que destacar para su uso en el Proyecto son los siguientes:

- 10 bits de resolución
- 13-260 μ s de tiempo de conversión
- Rango de tensión a la entrada desde 0 V hasta 5 V.

Con esto, lo primero que cabe pensar es que la señal de salida del circuito acondicionador descrito en el apartado 3.6.2. Sin embargo, por las limitaciones que ofrecen los amplificadores operacionales utilizados se va perdiendo tensión máxima a la salida. Esto podría corregirse mediante el uso de amplificadores operacionales *rail-to-rail*, que pueden ofrecer a sus salidas valores prácticamente iguales a los de su alimentación. Al colocar más amplificadores sin estas características, el rango de salida se va limitando y hace que saturate antes de lo previsto.

Con todo esto, en las entradas analógicas se recibirá una señal continua con una tensión que varíe entre los 0 V y aproximadamente los 3.1 V. Sabiendo que el convertidor tiene 10 bits y que el rango total que admite es de 5 V obtenemos lo siguiente:

$$\text{Resolución} = \frac{5 \text{ V}}{2^{10} \text{ niveles}} = \frac{5 \text{ V}}{1024 \text{ niveles}} = 4.883 \cdot 10^{-3} \frac{\text{V}}{\text{nivel}} = 4.883 \frac{\text{mV}}{\text{nivel}}$$

Esto implica que para identificar un cambio en la entrada es necesario una variación de 4.883 mV. Esta cantidad, si se va haciendo la transformación hacia atrás (desde la placa *Arduino*TM hasta el sensor de corriente), y teniendo en cuenta la diferencia entre valor de pico y valor eficaz, deriva en:

$$4.883 \frac{\text{mV}}{\text{nivel}} * \frac{1 \text{ V}}{4.17 \text{ V}} * \frac{1 \text{ V}}{1000 \text{ mV}} * \frac{30 \text{ A}}{1 \text{ V}} * \frac{1 \text{ A}_{RMS}}{\sqrt{2} \text{ A}} = 0.025 \text{ A}_{RMS}$$

Que, considerando que estamos conectados a la red con $V = 230 \text{ V}_{RMS}$ aproximadamente, conseguimos una precisión de unos 5 W (para cargas puramente resistivas). La lectura del pin de entrada analógica se verá suavizada por la salida de la etapa de detección de pico en el circuito acondicionador, y posteriormente por software, por lo que podremos obtener valores intermedios también.

De la placa también salen las conexiones necesarias para las comunicaciones con los demás módulos y componentes, que se irán detallando en los siguientes apartados.

3.6.4 Pantalla LCD

La pantalla es el elemento fundamental para la interacción del sistema con los usuarios. Se trata de un display de matriz de puntos sin retroiluminación del fabricante FORDATA, y muestra las siguientes informaciones:

- Al iniciar el sistema muestra un mensaje de encendido de todo el sistema.
- Tras este mensaje se muestra un mensaje de la inicialización del módulo de la tarjeta MicroSD. Si durante la inicialización se produce algún fallo, como la propia ausencia de la tarjeta, se muestra un mensaje de “ERROR SD”. De lo contrario, aparece un “OK.”, seguido de un mensaje que indica “Universidad de Sevilla”.
- Una vez inicializado todo se entra en el modo de funcionamiento normal, en el que en la primera línea se puede leer el intervalo de captura que está seleccionado en el momento. En la línea de abajo se actualiza a cada intervalo de captura el valor de la potencia activa en vatios calculada a partir del sensor de corriente.
- Si durante la escritura del archivo a la tarjeta en un ensayo se produjera un error en la tarjeta MicroSD o en el módulo del reloj RTC se mostraría también un mensaje de error indicando cuál es el módulo que ha fallado.
- Si durante el volcado de datos hacia el archivo fallase la propia operación de abrir el archivo, se muestra en la pantalla un mensaje indicando “Error al abrir el archivo”.



Ilustración 3.14. Display LCD

Con todas estas informaciones mostradas cualquier suceso interno en el sistema está bajo control.

La comunicación del display LCD con la placa de *Arduino*TM se hace mediante el bus I²C con lo que sólo hacen falta dos hilos para el envío de información. Para poder utilizar este protocolo ha sido necesario añadirle un módulo adicional que convierte la comunicación por I²C en comunicación en paralelo, ya que, como se puede ver en la **Ilustración 3.14**, lo que llega a la placa de la pantalla son 16 pines. Entre ellos se encuentran 8 pines dedicados a la recepción de información de forma paralela, lo cual podría conseguir una comunicación más rápida, pero para los fines de este proyecto no se requiere una gran velocidad. Para la comunicación en paralelo se pueden utilizar 4 de esos pines en lugar de los 8, ahorrando así la mitad de ellos. Aun así, para la comunicación por I²C tan sólo se requieren 2 pines digitales de la placa de *Arduino*TM (más los dos hilos de alimentación).

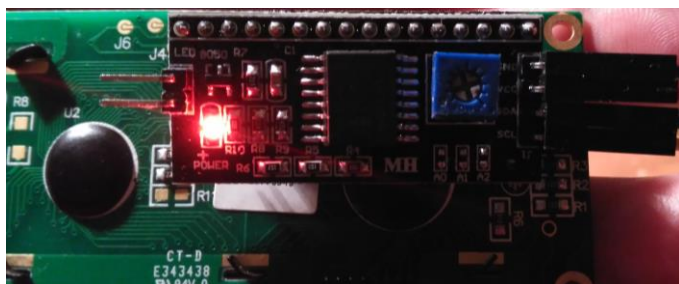


Ilustración 3.15. Interfaz I²C-Paralelo. El potenciómetro se usa para ajustar el contraste de la pantalla.

Como todo periférico conectado por vía I²C, este display LCD tiene asignada una dirección para su posterior manejo en el programa: 0x27.

3.6.5 Módulo de reloj RTC

Dado que la placa *Arduino*TM UNO no posee un reloj interno, es necesario añadirle uno externo para poder obtener la hora en el sistema y mantenerla una vez se le quite la energía. Éste es un pequeño módulo que incorpora un oscilador, una pequeña memoria RAM de 56 bytes y el circuito integrado DS1307, que es un estándar para los pequeños sistemas que requieren estar en fecha y hora. Este integrado también tiene en cuenta los días que tiene cada mes, incluidos los años bisiestos, así como tiene la posibilidad de proporcionar la hora en formato de 12 horas o 24 horas.

Para conseguir que mantenga la fecha y hora se le incorpora una batería de 3 V, que, según el fabricante, le puede proporcionar energía suficiente para conservarla durante más de 10 años.

En el *datasheet* especifica a qué zona de la memoria hay que acceder para conseguir los distintos campos de la fecha y la hora. Se detallará en el apartado del software.

Este módulo también utiliza el bus I²C para entablar la comunicación con la placa, por lo que no es necesario utilizar más pines del microcontrolador, sino que comparte los utilizados por el display LCD. Para ello se distingue por su dirección, especificada como 0x68.

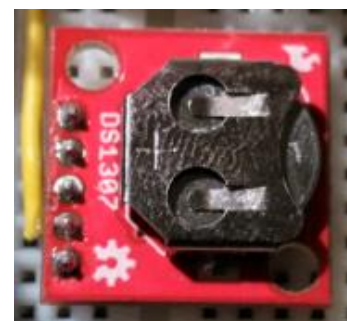


Ilustración 3.16. Módulo RTC

Aparte de los cuatro pines necesarios para la comunicación I²C tiene un quinto pin que genera una onda cuadrada programable, pero en este proyecto no se va a hacer uso de él.

El módulo RTC de Sparkfun, que es el que se ha utilizado, **requiere una inicialización** antes de instalarla en ningún sistema. Esto se realiza con una serie de comandos que sólo se ejecutará una vez, en los que se transfiere una fecha y hora de partida, a partir de la que seguirá contando hasta que se quede sin ninguna de las alimentaciones. Estas líneas quedan comentadas en el código del programa una vez establecida la fecha y hora.

3.6.6 Módulo MicroSD

En el último paso de este montaje, hay que pasar las medidas de potencia calculadas a un archivo y depositarlo en una tarjeta de memoria. Para este fin se añade al total este módulo del fabricante Catalex, que no tiene hoja de datos, pero funciona igual que un módulo idéntico de la marca Adafruit, que sí tiene. En la parte inferior indica cómo son las conexiones para la comunicación con él.



Ilustración 3.17. Módulo MicroSD. Vista superior.

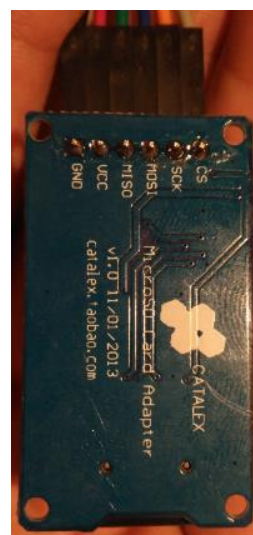


Ilustración 3.18. Módulo MicroSD. Vista inferior.

Como se puede deducir a partir de los nombres con los que se designa cada pin en la **Ilustración 3.18**, la comunicación empleada para este dispositivo es SPI. El porqué de usar este protocolo en lugar de I²C de nuevo es porque el segundo método se utiliza para periféricos con velocidades más bajas, es decir, que las operaciones de escritura y lectura en una tarjeta de memoria requieren mucha más velocidad que la que puede ofrecer el protocolo I²C². A cambio, requiere más conexiones físicas con el microcontrolador y, aunque no es el caso, cada módulo comunicado vía SPI con la placa requiere un hilo adicional por cada uno de ellos para seleccionarlo como *esclavo*.

3.6.7 Potenciómetro, pulsadores y diodos LED

Para la interacción entre el usuario y el dispositivo, se hace uso de dos pulsadores y un potenciómetro que cumplen distintas funciones. Los diodos LED tienen un código de luces con el que se identifican diferentes procesos.

El potenciómetro (resistencia variable de 20 k Ω) tiene como función ajustar el intervalo de captura de datos para mostrar y escribir en el archivo final. Girándolo en sentido horario pasa por cuatro regiones correspondientes a un intervalo de captura de 1 s, uno de 2s, otro de 5 s y un último de 10 s. De esta manera, el usuario puede programar este intervalo dependiendo de la duración prevista del ensayo. Esto evita que para ensayos muy largos haya grandes cantidades de datos muy constantes, dando como resultado un archivo de registro muy pesado y difícil de analizar.

Existen dos botones para indicar el comienzo y la finalización de un ensayo, es decir, el momento en que los datos ofrecidos en la pantalla LCD empiezan a volcarse en la tarjeta de memoria. Estos pulsadores están conectados a dos entradas digitales de la placa *Arduino*TM UNO.

² Mientras que la velocidad típica del bus I²C es típicamente de 100 kbps o de 400 kbps en su modo *ultrarrápido*, el bus SPI no tiene definida ninguna velocidad, por lo que puede trabajar a velocidades de 10 Mbps hasta del orden de 100 Mbps.

Por último, hay tres diodos LED que siguen un código de colores dependiendo de la función que se esté realizando en el momento:

- Un LED amarillo permanente indica que el sistema está encendido y tomando medidas.
- Un LED verde que se encenderá y quedará encendido permanentemente cuando se haya comenzado el registro de un ensayo.
- Un LED rojo que va pasando por diferentes etapas durante el programa
 - o al encender parpadea 5 veces muy rápido,
 - o durante la grabación de un ensayo efectúa un parpadeo por cada vez que se accede a la tarjeta de memoria y se vuelcan los datos correctamente,
 - o y en caso de error, por alguna de las razones descritas en el apartado **163.6.4**, se quedará encendido de forma permanente hasta que se solucione.

Con esto queda definido todo el hardware que envuelve al proyecto, dando como resultado un esquemático como el que sigue:

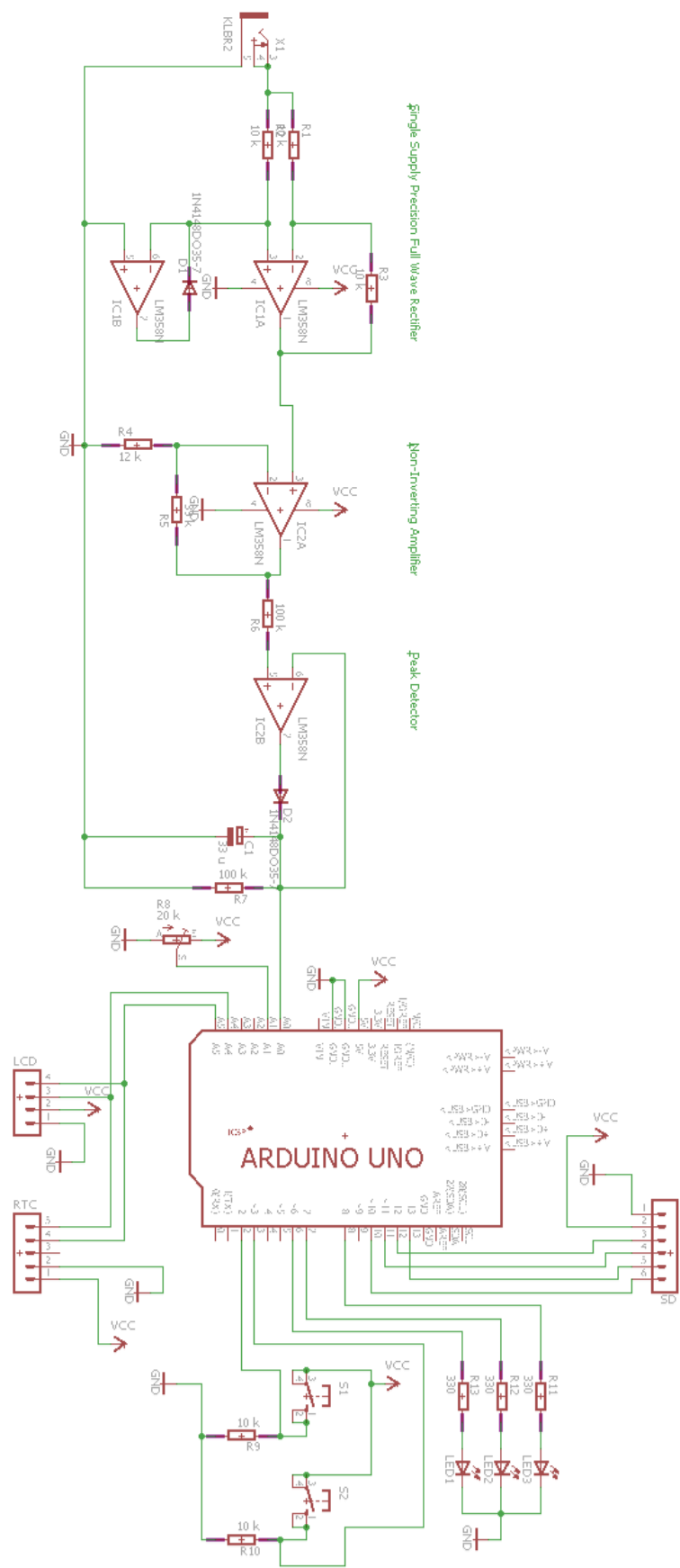


Ilustración 3.19. Esquemático final

A continuación se muestra más en detalle la parte del esquemático que se ha descrito desde el apartado 3.6.3:

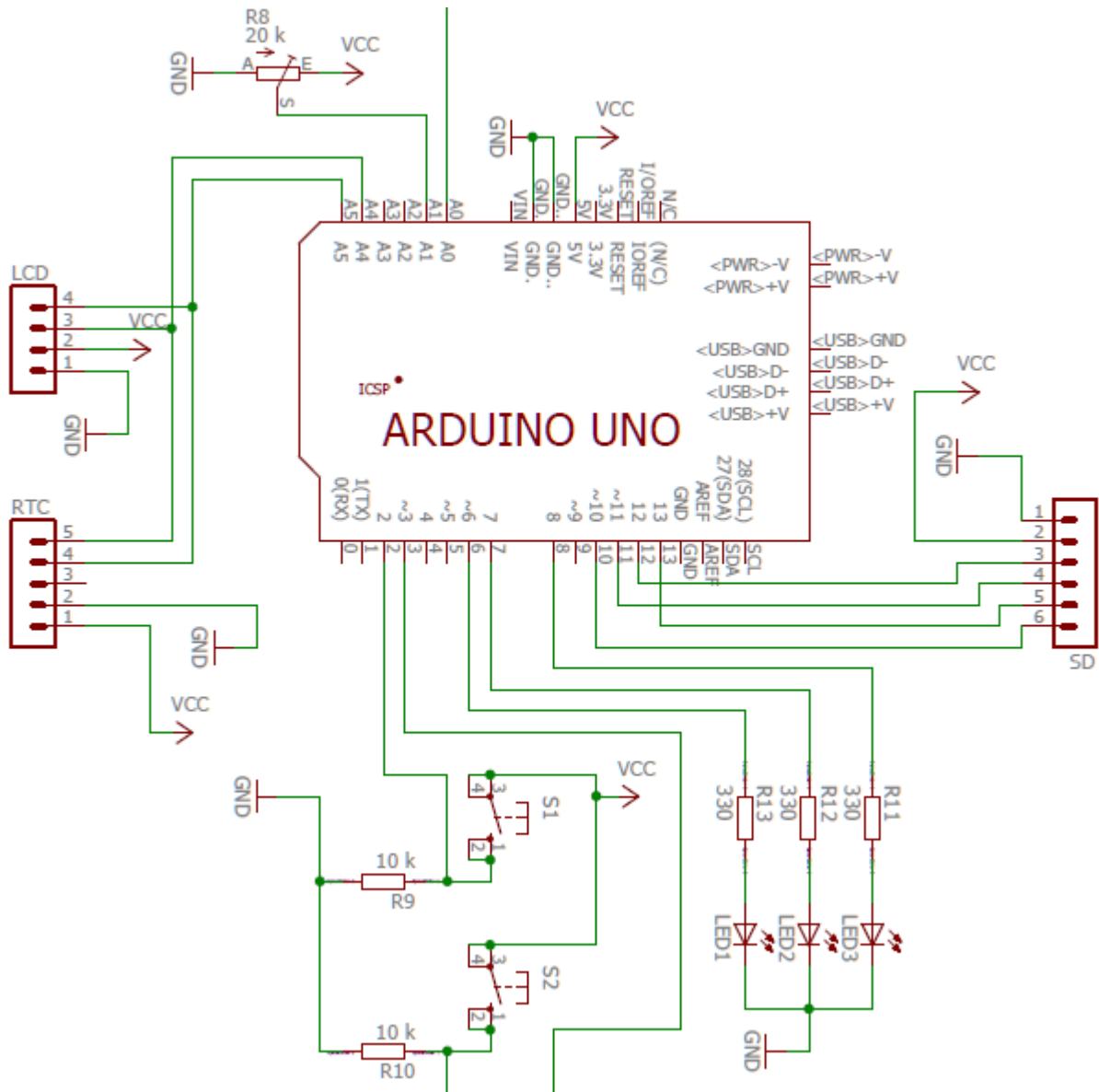


Ilustración 3.20. Interconexión de la placa *Arduino*TM con todos los componentes.

Donde la entrada designada como A0 es la salida del circuito acondicionador de señal del apartado 3.6.2.

3.6.8 El circuito de acondicionamiento de señal (versión inicial)

En la primera versión de este circuito, dado por un montaje de partida (situado en la bibliografía), se hacía pasar la señal del sensor por una primera etapa de rectificación y posteriormente una de amplificación regulable con un potenciómetro. El porqué de proceder de esta manera es fácil de entender y se explica a continuación, aunque incurría en imprecisiones desde el punto de vista hardware y (como se verá en siguientes secciones) software.

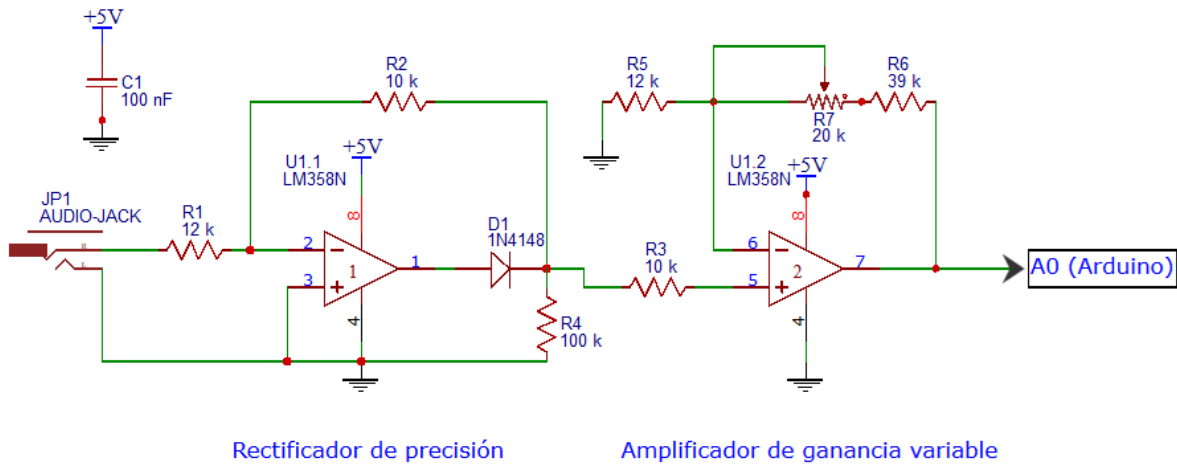


Ilustración 3.21. Esquemático del circuito de acondicionamiento de señal

Debido a la relación de transformación que se demostró previamente, la tensión obtenida a la salida del sensor de corriente es, en la mayor parte de su rango de funcionamiento, muy pequeña, sinusoidal y centrada en cero. De esta manera, y para que el cálculo de la potencia (hecho posteriormente por software) se pudiera realizar, se hacía una rectificación de onda completa, de forma que se pasa de una onda sinusoidal a una onda pulsante, como se aprecia en las imágenes del osciloscopio.

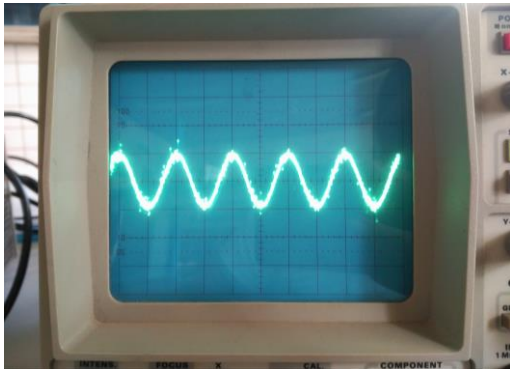


Ilustración 3.22. Señal del sensor antes de ser rectificada

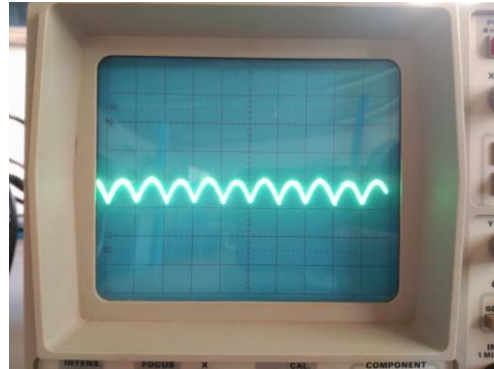


Ilustración 3.23. Señal del sensor después de ser rectificada

Se observa que se rectifica correctamente, en tanto a que se consigue la mitad de amplitud y el doble de frecuencia con la onda completamente rectificada.

Tras esta etapa se amplificaba la señal para aprovechar todo el rango de entrada posible de los pines de la placa *Arduino*TM UNO, según lo explicado en el apartado 3.6.3. Así, se amplificaba hasta la medida deseada mediante el potenciómetro designado como R₇, tal que para una salida del sensor de 1 V tengamos en el pin de entrada de *Arduino*TM aproximadamente 5 V.

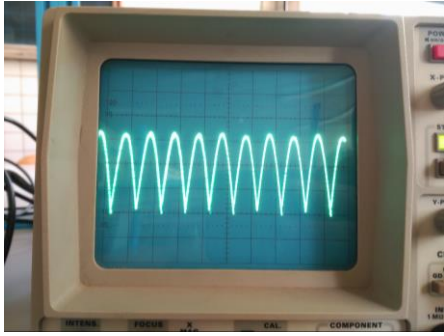


Ilustración 3.24. Señal del sensor amplificada

En la imagen de la izquierda se puede observar el resultado de la señal amplificada. Se muestra para fines meramente ilustrativos, ya que se ve a simple vista que no hay una relación 5:1 con respecto a la señal rectificada de la **Ilustración 3.23**.

Aunque lo cierto es que el circuito descrito cumplía con lo que estaba destinado a hacer, el análisis en cada punto del circuito no estaba muy bien definido y la necesidad o no de ciertos componentes utilizados no estaba del todo justificada, por lo que se pasó al montaje definitivo

3.7 Descripción software

En esta parte de la memoria se explicará cómo funciona el programa diseñado para el microcontrolador con el objetivo de cumplir con los distintos requisitos.

De manera análoga a la descripción por bloques, se presenta el siguiente diagrama de flujo

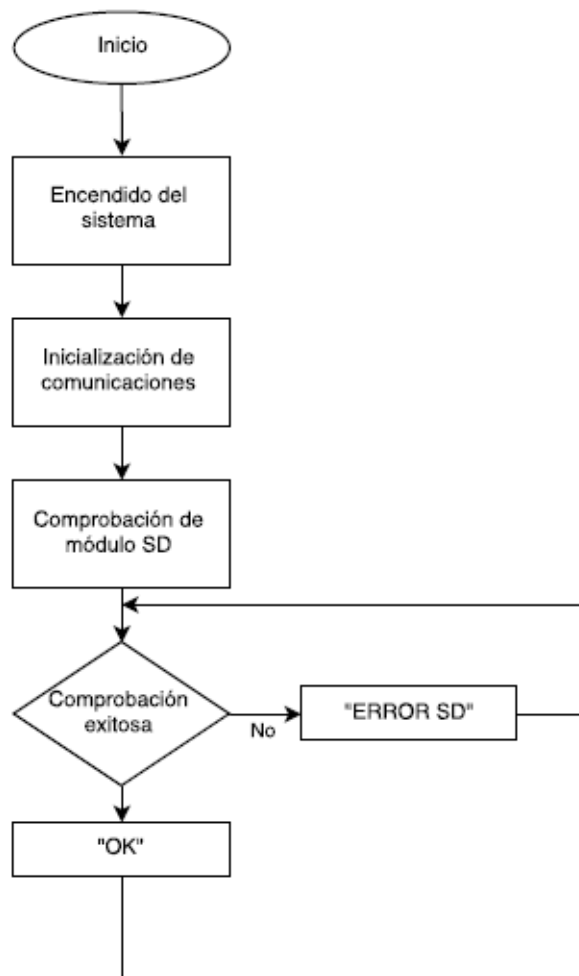


Ilustración 3.25. Primera parte del diagrama de flujo del programa

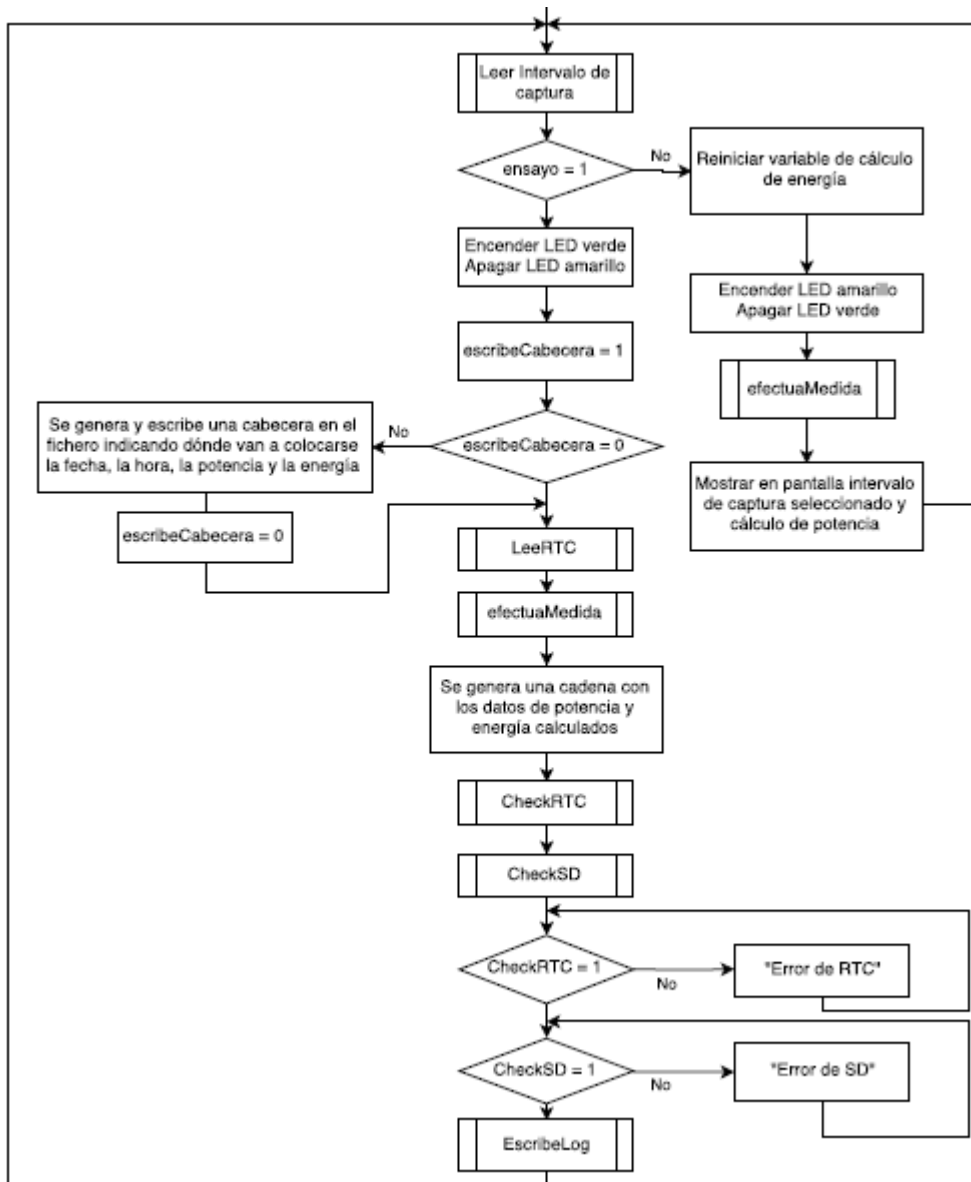


Ilustración 3.26. Segunda parte del diagrama de flujo del sistema

La primera parte del diagrama (**Ilustración 3.25**) se corresponde con la función `void setup()`, que es una función que se va a ejecutar siempre que se encienda el sistema y se inicie el programa. En esta función se albergan distintas órdenes de inicialización y comprobación de variables como se verá en el siguiente apartado.

La segunda parte del diagrama (**Ilustración 3.26**) se corresponde con la función `void loop()`, que es la segunda función que siempre se va a ejecutar cuando se encienda el sistema. Como su nombre en inglés indica, se trata de un bucle. Este bucle es infinito, por lo que se correspondería con un bucle `while(1)` tradicional en lenguaje C. Aquí se ejecutan cíclicamente todas las funciones necesarias para las que se ha desarrollado el programa, que se detallarán también a continuación.

3.7.1 Inicialización de las comunicaciones, periféricos y módulos

La unidad encargada de gestionar las comunicaciones con los periféricos de *Arduino*TM debe estar activada para establecer dichas comunicaciones. Para ello se emplean varias funciones:

`Wire.begin()`: inicializa el puerto de comunicaciones I²C.

`Serial.begin(9600)`: opcionalmente se puede inicializar la comunicación por el puerto serie de la placa en caso de que se quiera hacer un seguimiento de los procesos internos, con el ordenador conectado a la placa. El número entre paréntesis indica la velocidad de transmisión en baudios³.

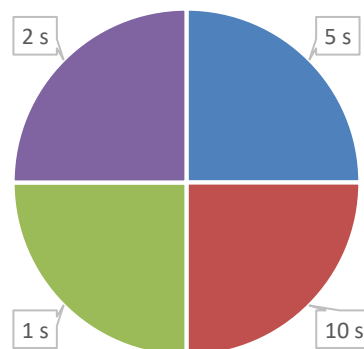
`lcd.init()`: se envían una serie de comandos vía I²C a la pantalla para que borre cualquier contenido que pudiera tener almacenado previamente y se deja en blanco.

`sd.begin(chipSelect, SPI_HALF_SPEED)`: con esta función se comprueba que la tarjeta de memoria está conectada y disponible para usar. Si todo es correcto, devuelve un 1. De lo contrario, 0.

3.7.2 Lectura del intervalo de captura

Como se describió en el apartado 3.6.7, hay un potenciómetro conectado a una de las entradas analógicas de la placa *Arduino*TM UNO. Al estar conectada en sus extremos a los valores de alimentación, devuelve una salida de tensión que será leída por el ADC de la placa. Esta lectura se efectúa mediante la función `void leerIntervalTime()` una función que no recibe argumento alguno ni devuelve nada, tan sólo modifica el valor de la variable global que indica el tiempo de captura de datos. Para ello, se hace uso de la `float averageRead(0, 50)`, donde el primer número indica el número del pin analógico y el segundo número la cantidad de muestras que se toman. Se toman 50 muestras (en este caso) de manera continua para hacer finalmente una media de todas y devolver un número (valor suavizado de la entrada) que se corresponderá a uno de los valores prefijados. El número tan alto de lecturas para suavizar la entrada se debe a que hay que evitar que el potenciómetro alterne entre dos valores que devuelvan salidas distintas aleatoriamente. Se ve en el siguiente esquema de forma gráfica

Posición del potenciómetro



Las lecturas de la potencia, tanto durante el ensayo como fuera de él, se realizarán a cada tanto tiempo como esté configurado esta variable según el potenciómetro. La energía consumida sólo se calcula durante el ensayo.

³ Un baudio es la cantidad de bloques de información que se envía por segundo, donde los bloques de información pueden contener uno o varios bits. Cuando cada bloque de información es un bit, entonces la velocidad en baudios coincide con la tasa de bits.

3.7.3 Comienzo y detención de ensayo

Para controlar cuándo empezar a grabar datos y cuándo parar, se hace uso de los pulsadores que se describieron con anterioridad. Éstos están conectados a dos pines digitales, de manera que activan las dos posibles interrupciones que admite la placa de *Arduino*TM UNO, mediante las funciones `void comenzarEnsayo()` y `void terminarEnsayo()`, que cambian el valor de la variable global `Ensayo` a 1 (para empezar) y a 0 (para acabar). La función `void comenzarEnsayo()` también modifica el valor de la variable `escribeCabecera` a 1 para generar la cadena de cabecera que distingue un ensayo de otro. Esta cadena es la siguiente:

```
“month;day;hour;minute;second;power;energy”
```

Así, al escribirlo en un archivo se pueden separar por columnas automáticamente.

Para activar las interrupciones es necesario que en la función `void setup()` se escriban las siguientes líneas:

```
attachInterrupt(digitalPinToInterrupt(2), comenzarEnsayo, RISING);  
attachInterrupt(digitalPinToInterrupt(3), terminarEnsayo, RISING);
```

Con las que se está especificando que en los pines 2 y 3 se están esperando interrupciones, que invocarán a las funciones correspondientes, y que lo harán cuando estos pines detecten un flanco de subida.

3.7.4 Comprobación de funcionamiento del módulo de reloj

Para comprobar que el reloj está disponible se hace uso de la función `int CheckRTC()`, que ejecuta una serie de comandos de la siguiente manera:

- Se genera un número aleatorio.
- Se guarda en una variable.
- Se empieza la comunicación con el módulo
- Se escribe el valor de esta variable en una posición de memoria del módulo RTC (0x09).
- Se acaba la comunicación, y se vuelve a iniciar
- Ahora se lee la información almacenada en la misma posición de memoria y se comprueba que coincide
- Si la lectura es igual que la escritura, entonces se considera que el módulo funciona correctamente y se devuelve un 1.
- En caso contrario, se devuelve un 0.

3.7.5 El cálculo de la potencia y la energía

La función con la que se va realizar el cálculo de la potencia activa consumida en el momento es `float efectuaMedida()`, que tiene la siguiente estructura:

- Al igual que en la lectura del potenciómetro, se lee la entrada correspondiente al sensor haciéndola pasar por la función `averageRead(1, 30)`, que suaviza la lectura tomando 30 muestras. En este caso se usan menos muestras que en la función de la lectura del potenciómetro porque la señal ya viene bien rectificadas y sin rizado, por lo que no es necesario un suavizado tan grande.
- Como la lectura es el valor de pico de la señal amplificada, se hace la conversión a valor eficaz (dividido entre $\sqrt{2}$) y se reduce por la ganancia (dividir entre 4.1).
- Ya que la entrada se está leyendo en una escala de 0 a 1023 niveles y con un rango total de 5 V se hace la conversión multiplicando por 5 y dividiendo por 1024.
- Por último se multiplica por el ratio que ofrece el sensor de corriente, que es 30, y se obtiene la corriente eficaz que está circulando por el cable que alimenta a la máquina.

- A partir de este valor calculado y el valor prefijado de la tensión de alimentación, se hace la multiplicación $P = U * I * \cos(\varphi)$ para obtener la potencia.
- Se separa el resultado de la potencia entre parte entera y dos decimales para su escritura en el archivo de destino. De esta manera se puede elegir entre separar los decimales con comas o con puntos, ya que dependiendo del equipo que vaya a procesar los datos posteriormente se deberá configurar de una manera o de otra.
- La energía consumida es un sumatorio de las energías consumidas hasta el momento, que se calculan multiplicando la potencia calculada por el intervalo de captura. Cuando acaba el ensayo, se resetea esta variable.

3.7.5.1 El cálculo de la potencia (versión inicial)

De la misma manera que el circuito de acondicionamiento, la función que calculaba la potencia activa consumida según la señal de entrada seguía una rutina distinta y más imprecisa, por lo que se modificó a esta versión final. En la versión inicial, la entrada era una señal sinusoidal rectificada y amplificada, pero sin una etapa de detección de pico, por lo que permanecía como se muestra en la **Ilustración 3.24**. La secuencia que se usaba entonces para hacer el cálculo era la siguiente:

- Esperar a que el ADC detecte un 0 de la señal.
- Una vez detectado, esperar 5 milisegundos hasta el pico más cercano.
- Tomar el valor.
- Repetir esta medida tantas veces como se especifique para hacer una media.
- Pasar el valor de pico a valor eficaz.

A partir de ahí la estructura era similar, pero cabe preguntarse: ¿el ADC va a detectar siempre un paso por 0 ante cualquier señal? Se ilustra mediante la siguiente imagen:

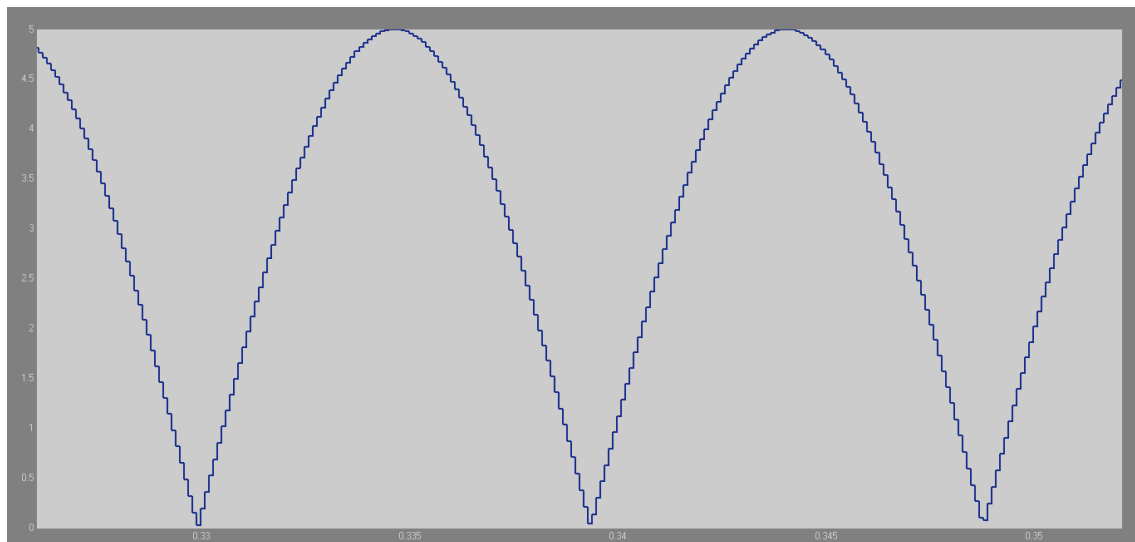


Ilustración 3.27. Emulación de una señal rectificada leída por el ADC de *Arduino*TM

Como se observa, el convertidor de la placa tiene una resolución y un tiempo de muestreo que limitados, y para valores altos de una onda de entrada el paso por cero tiene una pendiente muy acusada, por lo que en muchas ocasiones no detecta ese punto (como se aprecia arriba). Esto provocaba que el tiempo de adquisición de la medida fuese cuasi aleatorio, en tanto a que, una vez transcurrido el intervalo de captura, se tardaba un tiempo arbitrario de más hasta calcular un valor de potencia.

3.7.6 La lectura de fecha y hora

El módulo de reloj que se usa tiene integrado un chip DS1307, en cuyo *datasheet* se especifica cómo está almacenada la información, según el siguiente esquema:

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds			Seconds	00–59	
01h	0	10 Minutes			Minutes			Minutes	00–59	
02h	0	12	10 Hour	10 Hour	Hours			Hours	1–12 +AM/PM 00–23	
		24	PM/ AM							
03h	0	0	0	0	DAY			Day	01–07	
04h	0	0	10 Date		Date			Date	01–31	
05h	0	0	0	10 Month	Month			Month	01–12	
06h	10 Year				Year			Year	00–99	
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

0 = Always reads back as 0.

Ilustración 3.28. Esquema interno del módulo RTC

A través de esta tabla se van leyendo los diferentes bytes de la memoria del módulo, obteniendo, por orden, los segundos, minutos, horas (en formato de 12 horas o 24), día, día de la semana (del 1 al 7), mes y año. El byte siguiente se usa para la onda cuadrada opcional y de ahí en adelante es memoria interna, por lo que se detiene la solicitud de información en este punto. Esto se refleja en las siguientes líneas de código:

```
Wire.beginTransaction(DS1307_I2C_ADDRESS);
Wire.write((byte)0x00);
Wire.endTransmission();
Wire.requestFrom(DS1307_I2C_ADDRESS, 7);
second    = bcdToDec(Wire.read() & 0x7f);
minute    = bcdToDec(Wire.read());
hour      = bcdToDec(Wire.read() & 0x3f);
dayOfWeek = bcdToDec(Wire.read());
dayOfMonth = bcdToDec(Wire.read());
month     = bcdToDec(Wire.read());
year      = bcdToDec(Wire.read());
```

La función `bcdToDec` se encarga de convertir un dato cifrado en binario decimal (BCD) a decimal.

3.7.7 La escritura en el archivo

Para volcar los datos calculados a la tarjeta de memoria MicroSD se hace uso de la función `void EscribeLog(String dato)`, que recibe una cadena de caracteres generada con los valores de fecha, hora y valores calculados y accede a un archivo denominado `"Ensayo.csv"` para volcar esta cadena, donde `".csv"` se refiere a `"Comma-separated values"` (valores separados por comas). En nuestro caso están separados por el carácter `","` (punto y coma), pero el resultado es el mismo. Es por esto que hay que prestar especial atención a la forma en que se separan los decimales y se ofrece la posibilidad de hacerlo de una u otra manera.

Para hacer efectiva la escritura en el archivo `"Ensayo.csv"` se hace uso de la función `myFile.println(String dato)` donde `myFile` es una variable tipo `SdFile`. La información colocada en la cadena `dato` está ordenada según indica la cabecera según se indicó en el apartado 3.7.3.

3.7.8 Librerías usadas

Como se ha podido deducir de los apartados anteriores, se han usado funciones y comandos que no son de uso común y para lo cual hay que añadir librerías adicionales. En este caso, ha habido que usar tres, por lo que en las primeras líneas del código del programa se puede leer lo siguiente:

```
#include "Wire.h"           //Librería para las comunicaciones I2C
#include <SdFat.h>          //Librería para las comunicaciones SPI de la tarjeta
#include <LiquidCrystal_I2C.h> //Librería para el manejo del LCD por I2C
```

La primera librería, `Wire.h` es del propio entorno de *Arduino*TM, las otras dos han sido desarrolladas por usuarios y obtenidas a través de la plataforma de desarrollo software GitHub.

3.8 Diseño de PCB

Una vez que está determinado todo el hardware y el software de nuestro sistema, se diseña una placa de circuito impreso (PCB) mediante un programa de diseño. En el caso que nos ocupa, se ha usado la versión gratuita del software EAGLE 7.5.0. Light, con el que también se han diseñado los esquemáticos mostrados en apartados anteriores.

De esta manera, se hace un rutado a doble cara respetando unas condiciones de:

- Grosor mínimo de pistas
- Espaciado mínimo entre pistas
- Diámetro mínimo de taladro
- Establecimiento de un plano de masa en la cara de pistas

Teniendo en cuenta estas consideraciones, se colocan los elementos del circuito que tengan que estar en una posición determinada los primeros. Estos son: los pulsadores, el potenciómetro y los diodos LED. Tras esto, se hace el rutado de todo el circuito.

El diseño de la PCB se hará de manera que el circuito entero se sitúe encima de la placa de *Arduino*TM, encajando pin a pin (lo que se denomina “*shield*”) Esto se consigue gracias a la existencia, de nuevo, de usuarios que modelan la placa para poder usarla en el entorno de diseño de EAGLE.

Al terminar, se comprueba si hay algún tipo de error por incumplimiento de alguna de las condiciones anteriores, y cuando se han corregido se exporta finalmente una imagen a PDF para imprimir. En ese orden, tenemos las siguientes imágenes:

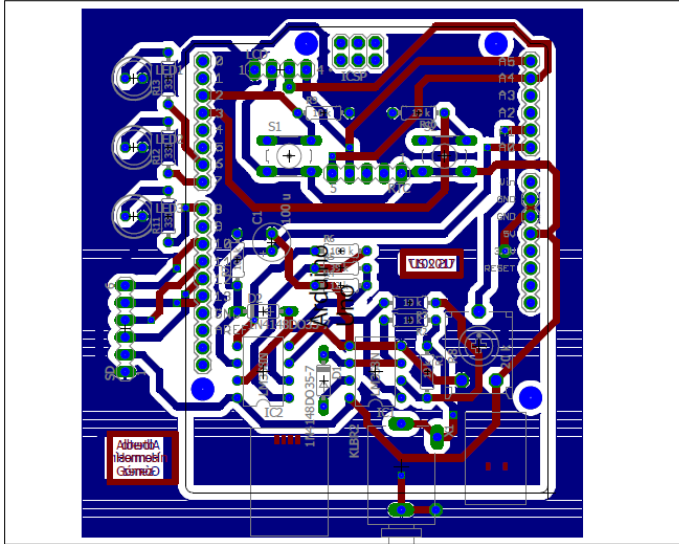


Ilustración 3.29. Vista superior de la placa con las dos caras superpuestas. En rojo la cara superior y en azul la inferior.

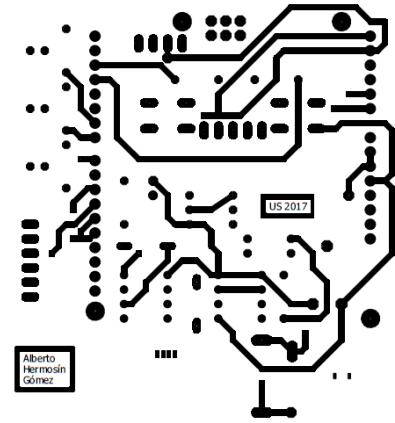


Ilustración 3.30. Fitolito de cara de componentes

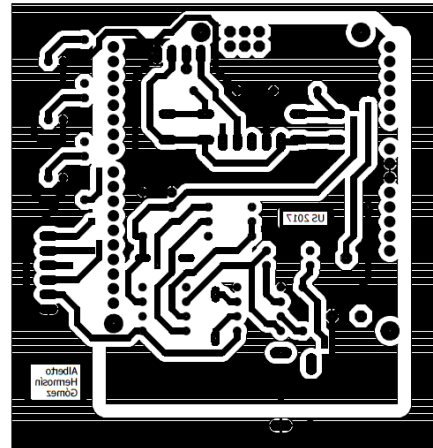


Ilustración 3.31. Fitolito de cara de pistas (reflejado)

3.9 Diseño de carcasa

Por último, y aprovechando las ventajas que la actualidad ofrece, se decidió que todo el sistema ya en placa de circuito impreso estuviera resguardado en una carcasa. Mediante el uso de un programa de dibujo 3D se ha llevado a cabo este proceso, resultando el siguiente diseño:

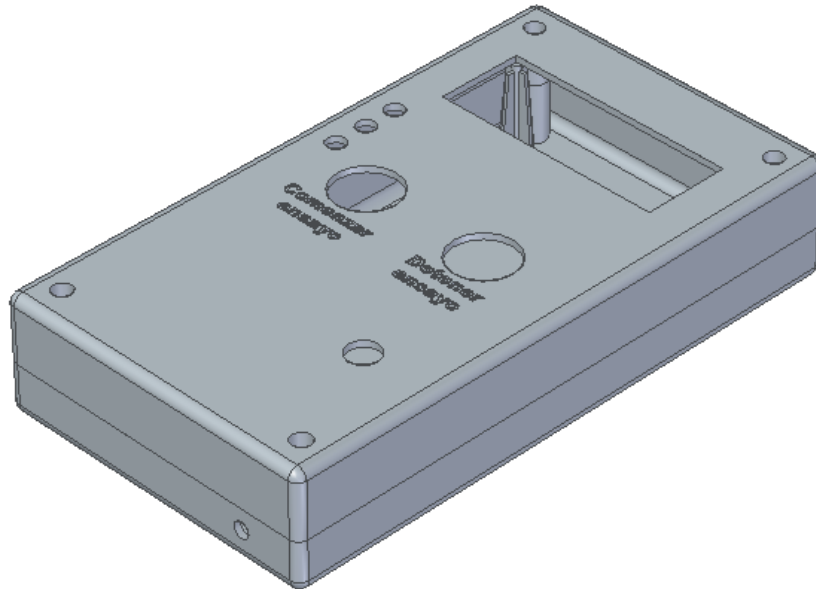


Ilustración 3.32. Vista general de la carcasa del sistema

En este diseño se puede comprobar cómo son accesibles los lugares como los botones para indicar el comienzo y detención de ensayos, el hueco para ajustar el potenciómetro y el espacio para la pantalla.

Además, en la parte inferior hay una perforación que permite la conexión del cable del sensor a la placa de forma cómoda.

Desde otro ángulo se pueden ver otras características importantes para el fácil manejo del dispositivo.

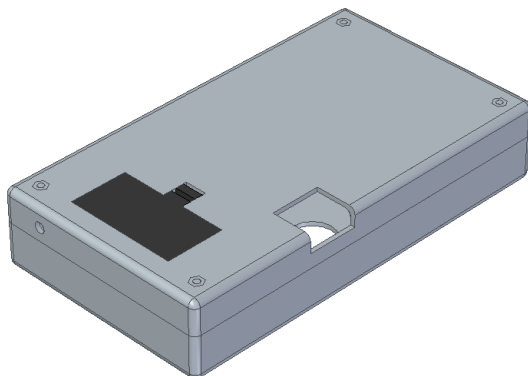


Ilustración 3.33. Vista inferior de la carcasa del sistema

En la parte trasera se incluyen dos elementos más. Por un lado, una ranura que permite la manipulación de la tarjeta de memoria en el dispositivo. Por otro lado, la tapadera negra se sitúa sobre una cavidad que otorga al sistema entero ser alimentado por una batería de 9 V. Aun así, y por si se prefiere alimentar por un adaptador de corriente, presenta una perforación en su interior de modo que esta conexión pueda ser posible.

De esta manera, se da un acabado final que cierra por completo todo el proyecto detallado en esta memoria.

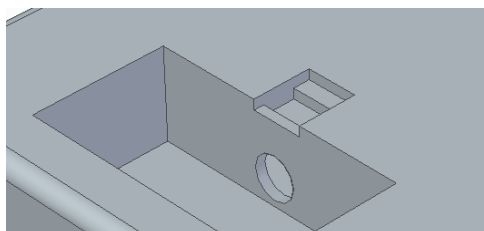


Ilustración 3.34. Detalle de la perforación para alimentación

4 INSTALACIÓN Y PUESTA EN MARCHA

Con el sistema completamente determinado, se pasa a instalar el sistema y realizar ensayos sobre la máquina y sus distintos subsistemas para comprobar que todo funciona según lo previsto y calculado. En este orden, se realizarán dos ensayos de unos cinco minutos de duración cada uno con una probeta de prueba, uno a cada subsistema de la máquina.

Para cada ensayo se necesita dar como dato al programa un factor de potencia, así que empezando por el primero, se programa como $\cos(\varphi_1) = 0.57$. Por otro lado, se mide la tensión a la que está conectado, que debe ser alrededor de $230 V_{RMS}$. Al medirlo se puede ver que está conectado a $226 V_{RMS}$. Por lo tanto el factor de potencia será ligeramente menor, . Para poder hacer la medición correctamente se procede de la misma manera que se mostraba en la **Ilustración 3.8**.

Programando el ensayo desde su propia interfaz del fabricante se le indica que se va a aplicar una carga cíclica de **10 kN**. Mirando en las especificaciones de la máquina de ensayos, encontramos la siguiente información:

RUMUL TESTRONIC at a glance						
Nominal Load	kN	50	100	150	250	
Max. static load	kN	50	100	150	150	250
Max. dynamic amplitude	kN	±25	±50	±75	±125	
Max. dynamic stroke	mm	4	6	5	4	
Frequency range ¹	Hz	approx. 40–260				
Frequency steps		8				
Dayligh between columns ²	mm	500				
Measurement and control accuracy static	%	<0,5				
Measurement and control accuracy dynamic	%	<0,5				
Vertical test space ²	mm	26–612			24–531	
Max. total height	mm	2.688			2.654	
Width	mm	690				
Depth	mm	1.010				
Total weight	kg	approx. 3.000				
Electric supply	V / Amp.	1 x 230 / 16			3 x 400 / 16	

Ilustración 4.1. Tabla de especificaciones de la máquina de ensayos de fatiga

Es decir, que para ensayos con cargas de hasta $100 kN$ se consumen hasta $16 A$ con conexión monofásica, que, recordando lo que especifica la placa de características, se traduce en un consumo de potencia de unos $2000 W$. Por tanto, para una carga 10 veces inferior a la mencionada, cabe esperar que el consumo de este conjunto sea de unas 10 veces menos.

4.1 Primer ensayo

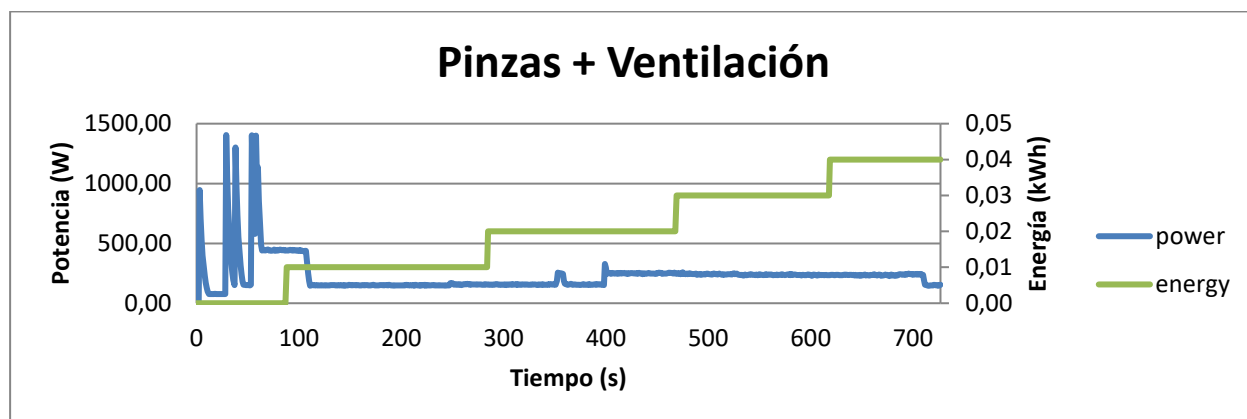


Ilustración 4.2. Representación del consumo eléctrico y la energía consumidas durante un ensayo de cinco minutos más otros cinco minutos previos al comienzo.

En este gráfico se pueden observar hasta diferentes etapas.

En la primera etapa (comienzo hasta 100 segundos aproximadamente) se observan picos altos de muy poca duración y regímenes permanentes por lo general superiores al resto de las etapas. Estos primeros picos se deben al manejo de las pinzas de la máquina, en sus movimientos de ascenso y de descenso. La explicación de por qué son consumos tan notorios es que el motor debe realizar un movimiento muy lento, controlado y constante mientras que debe **contrarrestar el peso del juego de pesas** de la caja superior (como se describió al principio en el apartado 1.2). Este elemento, que podía pasar desapercibido al principio, vemos que juega un papel importante en el consumo de la máquina si es necesario hacer muchos movimientos con las pinzas. Los picos altos resultan del arranque brusco del motor para mover las pinzas, que puede verse afectado por el esfuerzo de evitar la caída al vacío del juego de pesas que sostiene, debido a su inercia.

La segunda etapa (100 segundos hasta 400) es una etapa de reposo relativo, ya que tiene un consumo que se debe a que, desde el momento en que se pone en marcha la máquina, se activa un sistema de ventilación por aire, y que marca aproximadamente 150 W. Sin saber o no si es un valor de consumo válido o no, hace falta empezar el ensayo para ver el consumo cuando la máquina entra en acción. El pico que sucede en el segundo 360 aproximadamente puede deberse a algún tipo de proceso interno desconocido.

Esto se ve en la tercera y última etapa, donde se empiezan a generar los movimientos de tracción y compresión cíclicos (a una frecuencia de unos 120 Hz según indicaba la interfaz). El consumo se estabiliza al momento y permanece alrededor de los 230 W. Por tanto, la suposición del factor de potencia junto con la carga de una décima parte de la máxima parece corroborar que se está midiendo correctamente la potencia consumida en cada momento.

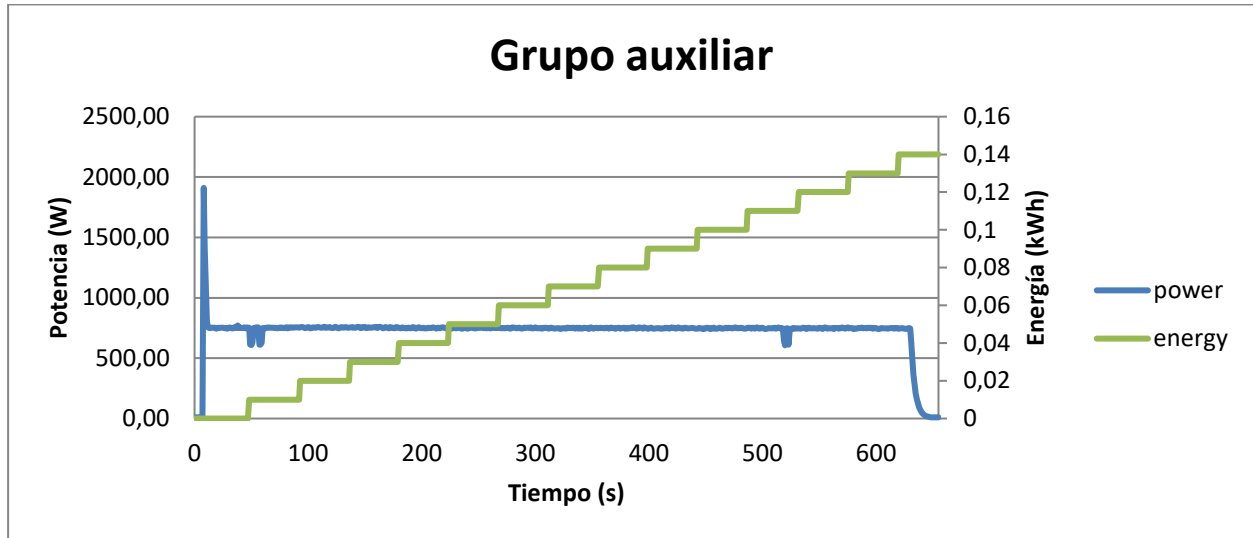
En línea verde se muestra el consumo de energía desde el comienzo del ensayo. Tiene una resolución relativamente baja ya que el volcado de datos se hace con dos decimales, por tanto, cambia con bastante lentitud. Aun así, se puede trazar una línea de tendencia donde se ve que es bastante lineal.

Cabe decir que, aunque no se ve reflejado en la gráfica anterior, cada dato capturado tiene su fecha y hora como se observa en esta otra imagen:

year	month	day	hour	minute	second	power	energy	
2017		9	12	16	49	15	13,80	0,00
2017		9	12	16	49	16	9,40	0,00
2017		9	12	16	49	17	9,40	0,00
2017		9	12	16	49	18	945,90	0,00
2017		9	12	16	49	19	705,80	0,00
2017		9	12	16	49	20	530,50	0,00
2017		9	12	16	49	21	398,60	0,00
2017		9	12	16	49	22	301,00	0,00

Ilustración 4.3. Imagen de la estructura del fichero generado

4.2 Segundo ensayo



En el segundo ensayo se observa una constancia absoluta en el consumo del grupo auxiliar (motor de presurización). Una vez puesto en marcha, presentando también un pico de menos de un segundo, se mantiene en su consumo nominal hasta detener la máquina.

El gráfico expuesto es una modificación a escala del obtenido mediante el monitor. En el resultado original, el valor constante de consumo se calculaba a través de un factor de potencia arbitrario. Tras ver el resultado de invariabilidad en todo el proceso del ensayo se determinó que este valor era el indicado en la hoja de características del fabricante, 750 W , por lo que se le introdujo un factor de potencia de 0.575.

Como se puede intuir en este punto, nuestro sistema no es capaz de computar un consumo generado por dos corrientes en distinta fase, ya que sólo le puede aplicar un factor de potencia a la medida tomada. Sin embargo, y a la luz de los resultados obtenidos en los ensayos, mientras que el consumo del subsistema principal es variable según los movimientos que haga, el subsistema de presurización del fluido tiene un consumo absolutamente constante desde el momento en que se enciende.

Tras este resultado podemos llegar a la siguiente conclusión que conllevará a una pequeña modificación en el código del programa:

El consumo del conjunto de los dos subsistemas se puede considerar como el consumo del subsistema principal más una cantidad fija de 750 W .

Dicho esto, el factor de potencia a computar en el código del programa será el calculado al principio, 0.57.

5 CONCLUSIONES Y POSIBLES AMPLIACIONES

Este monitor de consumo se ha probado capaz de medir cargas del orden de hasta 2000 W en pruebas, con un máximo proporcionado por el sensor de corriente de aproximadamente 3000 W para cargas puramente resistivas con una precisión muy aceptable, por tanto se podría extender su uso a casi cualquier aparato eléctrico, siempre que éste demande corriente en forma de onda sinusoidal y no con ninguna otra forma, como suele pasar en algunos cargadores de aparatos electrónicos, por ejemplo. Esto se debe al principio físico de funcionamiento del sensor, que ante una corriente constante no genera ninguna señal de salida.

Para terminar, habría que señalar algunas funcionalidades o características que podrían implementarse como ampliación de este proyecto.

5.1 Sistema remoto

Se estudia la posibilidad de instalarle algunos componentes que hagan posible su conexión remota con otros dispositivos que usen, por ejemplo, el sistema operativo de Android.

De esta manera se podría controlar lo que sucede a cada momento con el consumo de la máquina sin necesidad de estar físicamente, además de que se podrían programar los encendidos y apagados de manera más cómoda.

5.2 Generación de archivos

Hasta ahora, este sistema sólo es capaz de generar un archivo de texto de varias columnas separadas, en el que se van concatenando los diferentes ensayos mientras se desee. El límite de filas de Excel es suficiente como para almacenar datos calculados de casi dos semanas segundo a segundo. Aun así, una futura mejora sería la de generar un archivo con nombre variable que fuera, por ejemplo, el día y la hora a la que comienza el ensayo. En este sentido hay que destacar que los nombres de este tipo de archivos que podremos generar están muy limitados, sólo admiten 8 caracteres más los 3 caracteres de la extensión separados por un punto. Esto limita un poco la capacidad de nombrar los archivos, pero sería una mejora con respecto a lo actual.

GLOSARIO

ADC: Analog-to-Digital Converter	12
BCD: Binary-Coded Decimal	27
EAGLE: Easily Applicable Graphical Layout Editor	28
I ² C: Inter-Integrated Circuit	16
IDE: Integrated Development Environment	7
PCB: Printed Circuit Board	28
SBC: Single-Board Computer	7
SMD: Surface Mounting Device	11
SPI: Serial Peripheral Interface	17

BIBLIOGRAFÍA

Información sobre *Arduino*TM UNO: <https://store.arduino.cc/arduino-uno-rev3>

Ensayo de fatiga: <http://www.errekafasteningssolutions.com/en/fastening-solutions/fatigue-characterization/>

Montaje inicial: <http://www.diverteka.com/?p=1966>

Rectificador de precisión basado en amplificador operacional con alimentación simple:
<http://www.analogzoo.com/2016/01/single-supply-op-amp-rectifiers/>

Información del sensor SCT-013: [https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/files/Yhdc_CT_Report_\(Iss_6\).pdf](https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/files/Yhdc_CT_Report_(Iss_6).pdf)

Comunicación entre módulo Catalex MicroSD y Arduino:

<http://www.vishnumaiea.in/projects/hardware/interfacing-catalex-micro-sd-card-module>

Comunicación entre módulo RTC de Sparkfun y Arduino:

http://combustory.com/wiki/index.php/RTC1307_-_Real_Time_Clock

<http://www.glacialwanderer.com/hobbyrobotics/?p=12>

Adaptador de comunicación I²C a paralelo para el display: <https://fabricadigital.org/leccion/como-usar-el-modulo-i2c-con-una-pantalla-lcd-16022004/>

Comportamiento de un motor en el arranque: <http://www.ecmweb.com/content/evaluating-motor-and-transformer-inrush-currents>