



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO DE FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DISEÑO DE UN EQUIPO DE SUPERVISIÓN DEL FUNCIONAMIENTO DE UNA INSTALACIÓN ELÉCTRICA DOMÉSTICA.

AUTOR: CARLOS ANDREU TELLO

TUTOR: CARLOS ROLDÁN PORTA

COTUTOR: CARLOS ROLDÁN BLAY

Curso académico: 2016-17

AGRADECIMIENTOS

Me gustaría aprovechar la ocasión para agradecer el apoyo y la ayuda que he recibido de mis padres y hermana, compañeros y amigos. También me gustaría agradecer a todos los profesores que se han encargado de transmitir los conocimientos necesarios en cada clase, que han sido imprescindibles para llegar hasta el final. En especial quiero agradecer a mi tutor la ayuda recibida por su parte, tanto en clases como en tutorías y sobre todo durante la realización del TFG.

Muchas gracias.

RESUMEN

Este documento recoge todos los pasos que se han seguido para el diseño de un equipo capaz de supervisar una instalación eléctrica doméstica, la base del proyecto es el uso de un microcontrolador de la compañía Arduino, en especial el Arduino Uno Wifi.

Por lo tanto, a lo largo de este documento se encuentra detallado el hardware y software usado en el equipo de supervisión, fundamentos teóricos de su funcionamiento y los ensayos que se han realizado durante el proyecto, de forma que se pueda reproducir los pasos que se han seguido hasta encontrar la solución óptima.

Palabras clave: Detección electrodomésticos, actuación en electrodomésticos, intensidad, potencia aparente, microcontrolador Arduino, circuitos integrados.

ABSTRACT

This report comprises all the steps that have been followed to design an equipment capable to supervise a domestic electrical installation, the basis of the project is the use of a microcontroller from Arcuino Company, the microcontroller is Arduino One Wifi.

Therefore, this document details the specific hardware and software used on the equipment, the theoretical foundations on wich its operation relies, and the essays that have been performed during the product design, so that part could continue and replicate the steps followed to find the optimal solution.

Keywords: Detection home appliances, Acting in home appliances, Arduino's microcontroller, integrated circuits, current, Apparent power.

DOCUMENTOS CONTENIDOS EN EL TFG.

- MEMORIA.
- PRESUPUESTO.
- ANEXO 1. Esquemas de montaje
- ANEXO 2. Programación.

ÍNDICE DE LA MEMORIA

Capítulo 1. Introducción.....	10
1.1. Objetivo del documento.....	10
1.2. Motivación	10
1.3. Etapas	11
Capítulo 2. Antecedentes.....	12
2.1. Equipos de supervisión de instalaciones eléctricas	12
2.2. Microcontroladores.....	12
2.3. Arduino.....	13
Capítulo 3. Entorno de trabajo.....	14
3.1. Software.....	14
3.1.1. Arduino Ide	14
3.1.2. Processing.....	15
3.2. Hardware.....	16
3.2.1. Arduino Uno Wifi.....	16
3.2.2. Circuito integrado K3020P.....	18
3.2.3. Circuito integrado 4N25.....	19
3.2.4. Circuito integrado TL431.....	20
3.2.5. Circuito integrado W08DC.....	20
3.2.6. Protoboard.....	21
3.3. Laboratorio.....	21
Capítulo 4. Desarrollo.....	23
4.1. Ámbito de aplicación.....	23
4.2. Detección del estado de equipos que consumen energía eléctrica.....	23
4.2.1. Detección del estado de equipos mediante simulación con pines digitales 5V y GND.....	24
4.2.2. Detección del estado de equipos mediante puente de diodos y optoacoplador.....	26

4.3 Control del estado de equipos.....	27
4.3.1 Control mediante la placa Arduino Uno Wifi únicamente.....	27
4.3.2. Control mediante la placa Arduino Uno Wifi y optoacoplador fototriac.....	29
4.4 Medida de intensidad y cálculo de potencia aparente.....	31
4.4.1 Medida de una tensión continua.....	32
4.4.2 Medida de una tensión alterna utilizando el generador de ondas y la placa Arduino.....	34
4.4.3 Medida de intensidad.....	36
4.4.3.1. Cálculo de la constante para transformar tensión en intensidad.....	37
4.4.3.2. Medida de intensidad de tres bombillas en paralelo y cálculo de la potencia aparente....	37
4.4.3.3. Medida de intensidad de dos bombillas en paralelo y cálculo de la potencia aparente.....	39
4.4.3.4. Medida de intensidad de una bombilla y cálculo de la potencia aparente.....	40
4.4.3.5. Conclusiones de la medida de la intensidad y cálculo de la potencia aparente.....	41
Capítulo 5. Diseño Final.....	42
5.1. Funcionamiento del equipo.....	42
5.2. Alimentación del equipo.....	42
5.3. Montaje.....	45
Capítulo 6. Conclusiones.....	47
6.1. Posibles mejoras.....	47
6.2. Problemática encontrada al realizar el proyecto.....	47
6.3. Conclusiones.....	48
7. Catálogos.....	50
8. Bibliografía.....	51
9. Resumen de Figuras	52

ÍNDICE DEL PRESUPUESTO

1. Introducción.....	52
2. Contenido del Presupuesto.....	52
ANEXO 1. Esquemas de montaje	55
ANEXO 2. Programación.....	57

MEMORIA

Capítulo 1. Introducción.

1.1. Objetivo del documento.

El objetivo principal de este documento es elaborar una guía para el diseño de un equipo, el cual ha de ser capaz de obtener información acerca de los electrodomésticos o aparatos eléctricos que están consumiendo energía eléctrica, tener la capacidad de encender y apagar estos aparatos y dar información acerca del consumo energético. Este equipo estará constituido por una placa Arduino y diversos componentes electrónicos y eléctricos.

Para alcanzar esta meta previamente se han de cumplir los siguientes objetivos:

- En primer lugar, investigar la existencia de algún equipo en el mercado con similares características o que cumplan las mismas funciones, para posteriormente realizar una comparación.
- En segundo lugar, incrementar los conocimientos sobre microcontroladores, en especial de la marca Arduino, sobretodo en el aspecto de la programación y el análisis de los datos obtenidos, por otra parte, se busca aumentar los conocimientos acerca del entorno de desarrollo del programa Processing para realizar una interfaz gráfica.
- En tercer lugar, buscar y elegir los elementos que resulten interesantes para el desarrollo del equipo. Posteriormente, realizar pruebas que permitan escoger las mejores opciones para cada una de las funciones.

El objetivo final pero no el menos importante, será la aplicación de gran parte de los conocimientos adquiridos en cada una de las asignaturas del grado en ingeniería en Tecnologías Industriales. De manera particular, se aplicarán conceptos de electricidad, electrónica, programación, elaboración de proyectos.

1.2. Motivación.

Teniendo en cuenta que el ahorro energético tiene cada vez más importancia en nuestra sociedad, el objetivo del equipo que se va a desarrollar es ayudar a los usuarios a poder tener un ahorro energético de manera más sencilla, informando del estado de aparatos que pueden consumir energía

eléctrica, permitiendo tanto el encendido como el apagado de estos aparatos a distancia y mostrar la energía que están consumiendo.

Por otro lado, los microcontroladores poseen una capacidad muy elevada, es por esto que con el circuito adecuado se están utilizando en numerosas aplicaciones convirtiéndose en una alternativa económica para los usuarios. Sobre todo, en la industria, lo que provoca la necesidad de aprender a utilizarlos y a implementarlos.

Para el desarrollo del proyecto se van a utilizar los conocimientos aprendidos en el grado de ingeniería en tecnologías industriales, especialmente los conocimientos impartidos en materias de informática, electricidad y electrónica.

1.3. Etapas.

Para el desarrollo del proyecto se ha seguido el siguiente plan de trabajo:

- A. Búsqueda de los distintos tipos de microcontroladores de la marca Arduino en cuanto a características y funciones para encontrar el microcontrolador idóneo para el proyecto.
- B. Investigación y adquisición de los elementos que se necesitarán para llevar a cabo el proyecto.
- C. (Entendimiento y aprendizaje del lenguaje de programación, basado sobre todo en C++, utilizando tanto libros como portales web).
- D. Realización de diferentes sketches con las funciones individuales del proyecto.
- E. Desarrollo de los montajes y prueba del correcto funcionamiento de los sketches.
- F. Familiarización con el programa Processing, basado en Java, para la realización de una interfaz gráfica para el proyecto.
- G. Desarrollo de diferentes interfaces gráficas para los respectivos sketches.
- H. Realización de un sketch único que comprenda cada una de las funciones del proyecto.
- I. Prueba del correcto funcionamiento del sketch.
- J. Desarrollo de una interfaz gráfica única para el sketch.
- K. Prueba del funcionamiento tanto del sketch como de la interfaz gráfica.

La mayoría de los pasos han sido iterativos debido a que se han modificado tanto los sketches como las interfaces gráficas con el objetivo de optimizar el proyecto y conseguir un perfecto funcionamiento.

Capítulo 2. Antecedentes.

En este capítulo se va a revisar la existencia de distintos equipos de supervisión del funcionamiento de una instalación eléctrica y microcontroladores, ya que va a suponer la base del desarrollo del equipo de supervisión que se va a desarrollar.

2.1. Equipos de supervisión de instalaciones eléctricas

Debido al continuo aumento del precio de la energía eléctrica cada día es más importante controlar el suministro de energía de una instalación. Es por esto que se ha buscado en el mercado la existencia de equipos que sean capaces de realizar las siguientes funciones:

- Detección de electrodomésticos que consuman energía en una instalación eléctrica.
- Informar al usuario acerca del consumo de los electrodomésticos.
- Actuar sobre los electrodomésticos apagándolos o encendiéndolos.

No se ha encontrado en el mercado ningún equipo que sea capaz de realizar las anteriores funciones. No obstante, si se ha encontrado equipos que solo son capaces de actuar sobre los electrodomésticos como pueden ser la domotización de viviendas.

2.2. Microcontroladores.

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos. Están basados en los microprocesadores, aunque hay muchas diferencias entre ellos:

- Memoria. Estando integrada en el microcontrolador dentro del circuito integrado, mientras que en el procesador mientras que en el microprocesador son dispositivos externos que lo complementan para un funcionamiento óptimo
- Velocidad de operación. En comparación es más rápida la de un microprocesador que la de un microcontrolador.
- Tamaño. Teniendo en cuenta que el microcontrolador tiene todos los elementos en un circuito integrado provoca la disminución en el tamaño del circuito impreso por la reducción de los circuitos externos, por lo que el microcontrolador tendrá un menor tamaño que un microprocesador.

- Interferencias. El microprocesador es más susceptible a la interferencia electromagnética debido a su tamaño y a su cableado externo que lo hace más propenso al ruido. Por otro lado, como el microcontrolador tiene una alta capacidad de integración reduce los niveles de interferencia electromagnética.
- Tiempo de desarrollo. Es mucho más rápido en un microcontrolador que en un microprocesador.
- Precio. Los sistemas basados en microprocesadores son mucho más elevados que los sistemas basados en microcontroladores.

Los microcontroladores tienen un amplio campo de aplicación entre los que se destacan:

- Control de periféricos y dispositivos auxiliares en la industria informática.
- Electrodomésticos.
- Industria de automoción.
- Control industrial y robótica.
- Domótica.

2.3 Arduino.

Arduino es una compañía de prototipos de código abierto basada en hardware y software fáciles de usar. Para ello usa su propio lenguaje de programación Arduino (basado en el lenguaje Wiring) y el software propio de Arduino (IDE) , basado en el software Processing .

Arduino ha sido muy utilizado para muchos proyectos y ha conseguido crear una comunidad de usuarios (*makers*) que ofrecen sus códigos abiertos (*sketchs*) a la plataforma, que añadido a la colaboración entre los usuarios ha ayudado a crear un gran conocimiento accesible para cualquier persona que quiera realizar cualquier proyecto.

Según su cofundador *Massimo Banzi* (2011), el triunfo de Arduino está basado en:

Bajo coste comparado con otras plataformas de microcontroladores.

Compatibilidad de plataforma, ya que funciona tanto con Windows como con Macintosh, OSX y Linux a diferencia de otras plataformas que solo funcionan con Windows.

Entorno de programación simple y claro. El software de Arduino (IDE) es fácil de utilizar para los iniciados a la vez que es muy flexible para los usuarios más avanzados.

Código abierto y hardware extensible. En el mercado existe una gran variedad de shields distintas, pero gracias a que los planos de las placas están publicados, los diseñadores pueden extender y mejorar la misma añadiendo complementos (sensores, puertos de comunicación, relojes, entre otros) que permite aumentar las funciones de estas placas.

Capítulo 3. Entorno de trabajo.

El contenido de este capítulo se basa en conocer el software, hardware y el laboratorio que se ha utilizado para la realización del equipo de supervisión, de manera que quede definido todo el entorno de trabajo en el que se realizarán los ensayos.

3.1. Software.

Para desarrollar el equipo de supervisión se ha utilizado dos programas distintos Arduino Ide y Processing.

3.1.1. Arduino Ide

El programa Arduino Ide es el entorno de desarrollo de los sketches (código de programas de la placa), se puede descargar gratuitamente de la página web de Arduino para cualquier plataforma (Windows, Mac OSX, Linux). El entorno de programación está basado en *Processing* y el lenguaje en *Wiring*, compartiendo un gran número de funciones con los lenguajes *Java* y *C++*

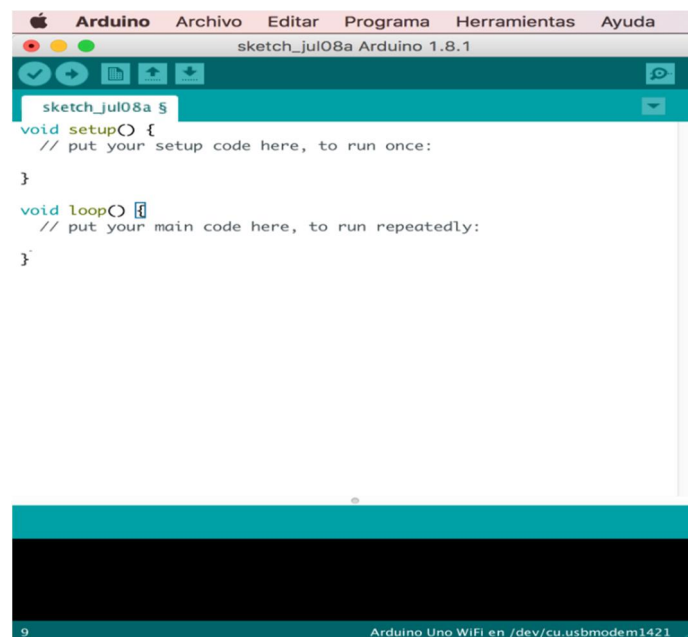


Figura 1. Pantalla programa Arduino Ide.

Como se puede observar en la figura 1, el entorno de programación se divide en cuatro partes. La primera parte es la barra del menú principal que se divide en seis pestañas (Arduino, archivo, editar, programa herramientas y ayuda). De las cuales se destacan las siguientes:

- Archivo, pestaña en la cual se pueden abrir tanto nuevos sketches, sketches guardados anteriormente y una serie de ejemplos básicos que ayudan a comprender tanto el funcionamiento de los sketches como el lenguaje de programación.
- Programa, que permite compilar el sketch, subir el sketch a la placa o incluir bibliotecas que no se encontraban en el programa al descargarse.
- Herramientas, en la que se debe configurar el tipo de placa, Arduino Wifi Uno y el puerto a través del cual se comunicará con el PC, puerto serie. Otra acción que se puede ejecutar en esta pestaña es abrir el monitor serie el cual nos permite enviar y recibir fácilmente información a través del puerto serie.

La segunda parte consta de una barra horizontal con seis botones que son accesos directos a las siguientes funciones respectivamente:

- Verificar/compilar
- Subir
- Nuevo
- Abrir
- Salvar
- Monitor serie

La siguiente parte se trata de la pantalla principal en la que aparecen dos funciones, la primera, *void setup ()*, donde se configurarán los parámetros de la placa como la velocidad de comunicación entre la placa y el PC, esta parte de la función solo se leerá una vez. La siguiente función, *void loop()*, es la función que lee las instrucciones de manera cíclica.

La última parte es una pantalla cuyo fondo es de color negro, esta pantalla sirve para informar el estado de los procesos de verificar/compilar y subir, así como suministrar información acerca del tamaño que ocupa el sketch dentro de la placa.

Es importante destacar que el programa contiene un gran número de bibliotecas, que se deben incluir al principio de empezar el sketch. Muchas bibliotecas se encuentran preinstaladas en el programa. Por otro lado, existe un gran número de bibliotecas que se han de instalar desde internet. La funcionalidad de las bibliotecas es aumentar las funciones de programación, son especialmente importantes en el caso de usar *shields* o comunicarse con otros programas.

3.1.2. Processing.

El programa *Processing* se ha utilizado para realizar una interfaz gráfica que permita visualizar las acciones realizadas por la placa. Este programa puede descargarse de manera gratuita desde su propia página web en diferentes plataformas (Windows, Mac OSX y Linux).

El programa se divide en tres partes como se puede observar en la figura 2. La primera parte es un conjunto de pestañas en las que se encuentran las funciones del programa. La segunda parte es la pantalla principal, una pantalla blanca en la que se introducen las instrucciones del sketch y dos botones, que sirven para reproducir y parar el sketch respectivamente. La tercera parte es una consola de comunicación que se utiliza para informar acerca de errores en el código.

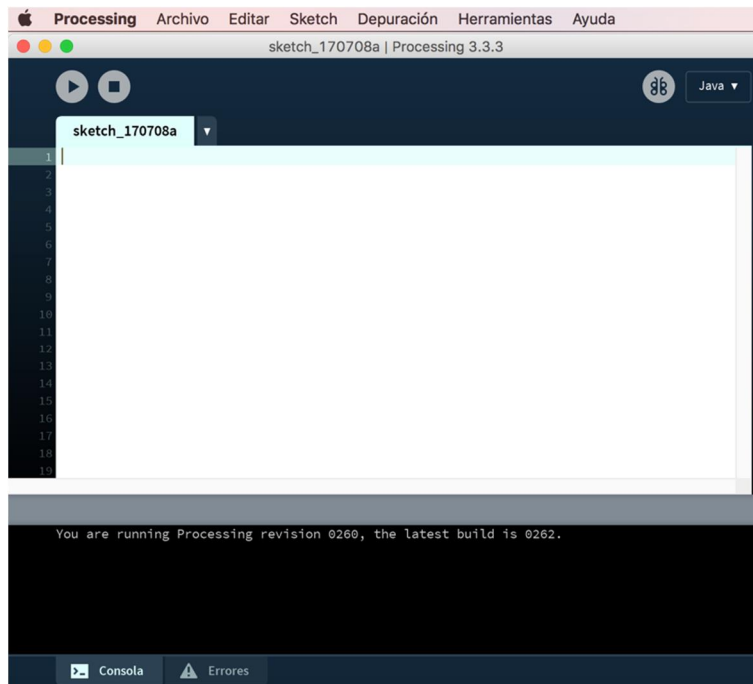


Figura 2. Pantalla del entorno de programación de *Processing*.

3.2. Hardware.

En este apartado se van a describir de manera resumida los elementos que han formado parte del hardware en el desarrollo del equipo, con el propósito de entender su función dentro del proyecto.

Todas las características de estos elementos se encuentran detalladas en los catálogos citados al final del texto.

3.2.1. Arduino Uno Wifi.

Arduino Uno Wifi es la placa que se va a utilizar en el equipo, esta placa es similar a la placa Arduino Uno Rev3 con un módulo Wifi integrado, la programación es idéntica al resto de placas, pero la tener un módulo Wifi integrado es posible subir el sketch a la placa de manera inalámbrica.

Esta placa tiene una dimensión de 53x68.5 mm y tiene un peso de 25 g. En la figura 3 se puede ver la placa Arduino Uno Wifi, con sus pins de entrada y salida (E/S) tanto analógicos como digitales, puertos, chips, etc.



Figura 3. Arduino Uno Wifi. Recuperado de <http://www.arduino.cc>

Las características más destacables son las siguientes:

- Alimentación:

Arduino Uno Wi-Fi se puede alimentar a través de la conexión USB (5V) o con una fuente de alimentación externa. La alimentación externa puede venir desde un adaptador AC-DC o una batería. La fuente de alimentación se selecciona automáticamente.

- Memoria:

El microcontrolador ATmega328 tiene 32 KB de memoria flash para guardar el código, 2 KB de SRAM y 1 KB de EEPROM.

- Entradas y salidas:

Cada uno de los 14 pines digitales se puede utilizar como entrada o salida, usando distintas funciones de programación (`pinMode ()`, `digitalWrite ()` y `digitalRead ()`).

Cada pin funciona a 5 voltios y puede proporcionar o recibir un máximo de 40 mA.

Por otro lado, Arduino Uno Wifi dispone de 6 entradas analógicas con 10 bits de resolución cada una (1024 valores distintos), estos pines tienen un rango de 0-5V por defecto, aunque es posible variarlo usando el pin Aref o con distintas funciones.

- Comunicación:

Arduino Uno Wi-Fi tiene una serie de opciones para comunicarse con un PC, otro Arduino, u otros microcontroladores. La primera opción es mediante el puerto serie, ya sea mediante conexión USB o a través de los pines digitales 0 (RX) o 1(TX). La segunda opción es mediante

los puertos TWI y SPI, los cuales gracias a la ayuda de las librerías facilitan su uso. La última opción es mediante una conexión Wifi, conectando la placa a la misma red Wifi que dispositivo desde el que se quiere controlar.

Por otro lado, como se ha descrito antes, el software de Arduino posee un monitor serie que permite mandar y recibir información.

A continuación, se incluye una tabla con las características más importantes tanto del microcontrolador como del microprocesador del Arduino Uno Wifi

Microcontrolador	ATmega328
Voltaje de operación (microcontrolador)	5V
Pines digitales (I/O)	20
Entradas analógicas	6
Velocidad de reloj	16 MHz
Memoria flash	32 KB
SRAM	2 KB
EEPROM	1 KB
Microprocesador	ESP8266
Voltaje de operación (microprocesador)	3.3 V
Wifi	802.11 b/g/n 2.4GHz
Velocidad de reloj (microprocesador)	80 MHz

Tabla 1. Principales características Arduino Uno Wifi.

3.2.2. Circuito integrado K3020P.

El K3020P es un optoacoplador fototriac. Consta de un fototriac acoplado ópticamente a un diodo emisor de infrarrojos. Su función es permitir el paso de corriente entre el diodo que tiene acoplado entre las patas 1 y 2 si circula intensidad entre las patas 4 y 6.

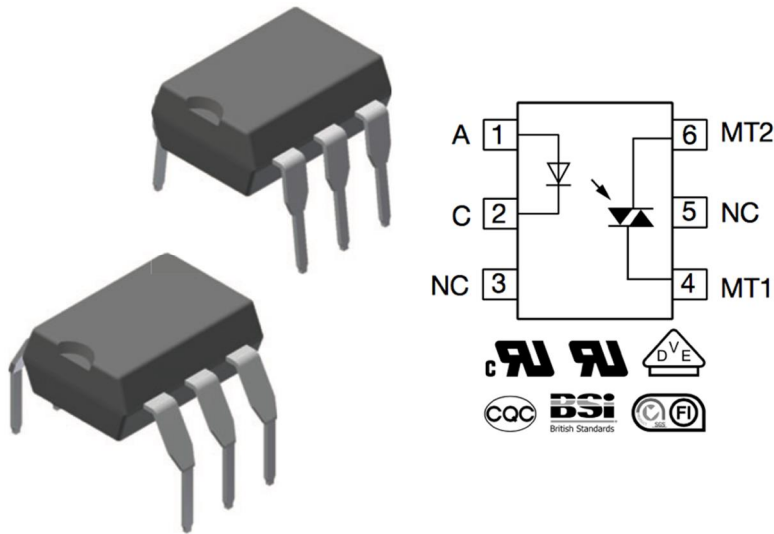


Figura 4. Imagen y esquema del circuito integrado K3020P. Recuperado del catálogo Vishay.

3.2.3. Circuito integrado 4N25.

El circuito integrado 4N25 es un optoacoplador, un optoacoplador es un circuito integrado muy básico compuesto

generalmente por un diodo LED y un fototransistor unidos de tal forma que cuando una señal eléctrica circula a través del LED haciendo que brille, la luz emitida es recibida por la base del fototransistor, que empieza a actuar en modo saturación.

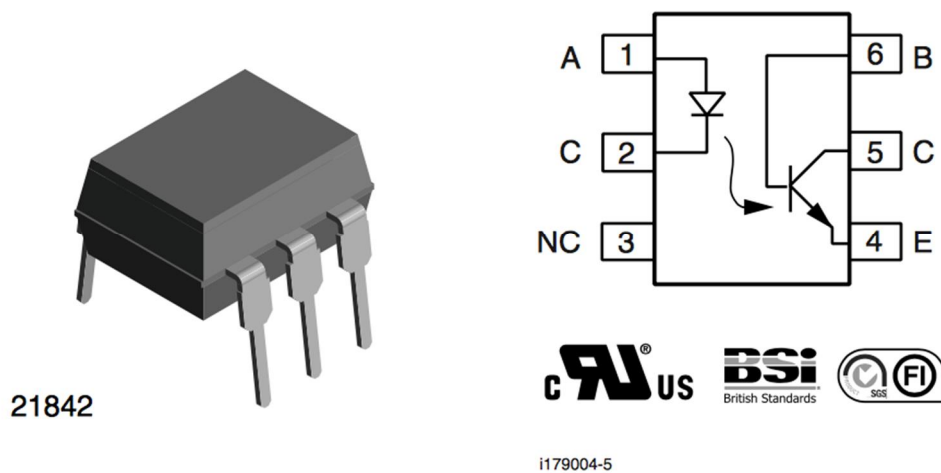


Figura 5. Imagen y esquema circuito integrado 4N25. Recuperado del catálogo de Vishay

3.2.4. Circuito integrado TL431.

El circuito integrado TL431 es una referencia de tensión regulable, entre 2,5-40 V, en función de dos resistencias externas. Su funcionamiento es como un diodo zener pero más estable y con mayor precisión, ya que garantiza la tensión ante diferencias de la intensidad suministrada y cambios de temperatura. Este componente se encargará de centrar las tensiones en el valor que se elija y no en cero, lo que facilitará que puedan ser leídas por el Arduino uno Wifi.

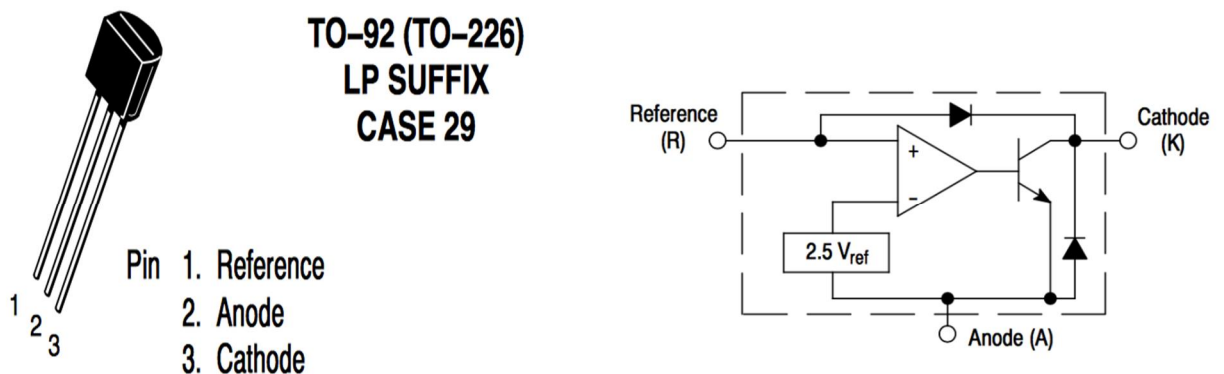


Figura 6. Imagen TL431y diagrama de bloque funcional. Recuperado del catálogo ON semiconductor.

3.2.5. Circuito integrado W08M.

El circuito integrado W08M es un puente rectificador de diodos. Un puente rectificador de diodos, también llamado puente rectificador o puente de diodos, es un dispositivo formado por cuatro diodos ensamblados de forma que una corriente alterna (AC) conectada a dos de los diodos produce una corriente continua (DC) de salida en los dos diodos restantes.

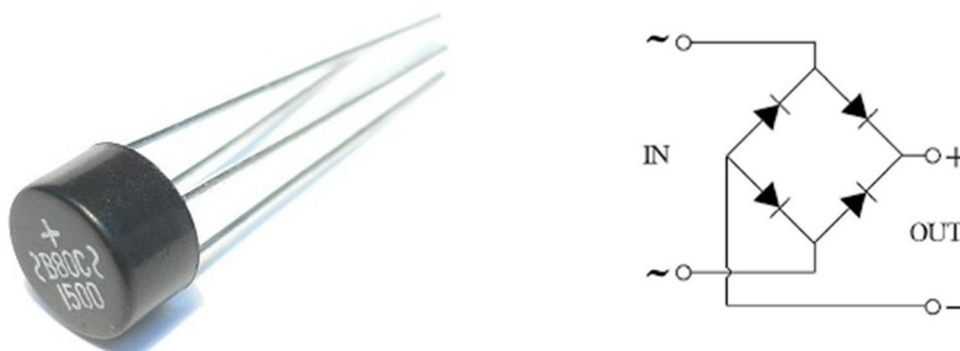


Figura 7. Imagen y esquema puente rectificador de diodos. Recuperado del catálogo de Futurlec

3.2.6. Protoboard.

Un protoboard es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y realizar prototipos de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí. Uno de sus usos principales es la creación y comprobación de prototipos de circuitos electrónicos antes de llegar a la impresión mecánica del circuito en sistemas de producción comercial.

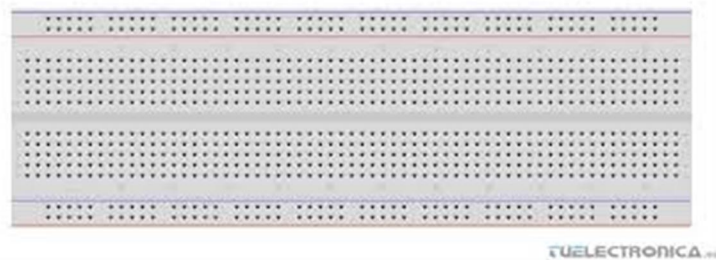


Figura 8. Protoboard. Recuperado de <http://www.cetronic.es>

3.3. Laboratorio.

Las pruebas se han realizado en el laboratorio de electricidad ubicado en el edificio 5E, para las pruebas del equipo se han utilizado un osciloscopio, un generador de ondas y un multímetro.

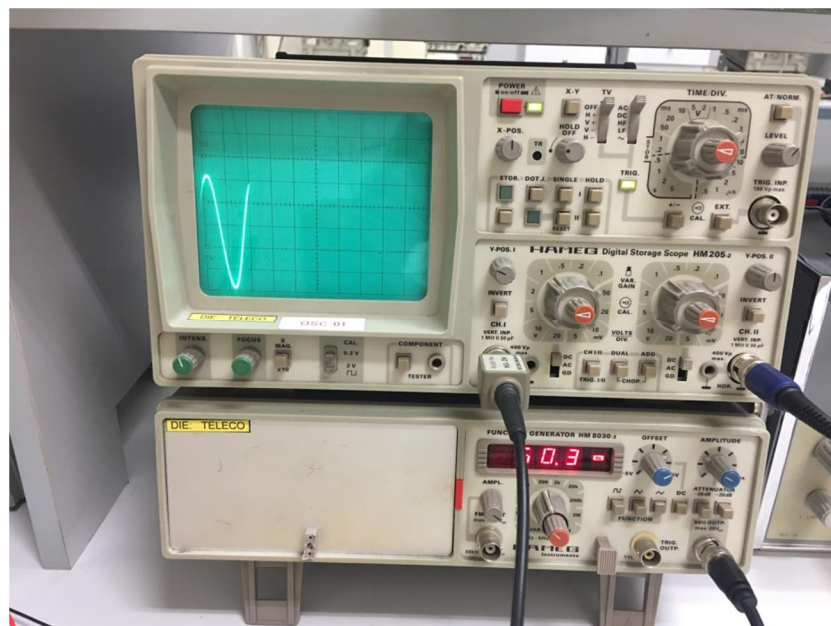


Figura 9. Osciloscopio y Generador de ondas



Figura 10. Multímetro

Capítulo 4. DESARROLLO.

En este capítulo se va a incluir cada uno de los pasos seguidos para la construcción del equipo, incluyendo las condiciones y esquemas utilizados para realizar los ensayos.

Para explicar este punto y en función de cómo se ha desarrollado el prototipo del equipo final se va a dividir en tantas partes como funciones se busca que cumpla el equipo de supervisión:

- Detección del estado de equipos que consuman energía eléctrica.
- Actuar sobre equipos que consuman energía eléctrica.
- Medir el consumo de estos equipos.

4.1 Ámbito de aplicación.

La idea inicial es diseñar un prototipo que supervise una instalación pequeña, esta instalación se basa en una habitación que posee 4 electrodomésticos, que simularán los equipos que consumen energía eléctrica y la iluminación de dicha habitación.

El equipo se ha diseñado para instalaciones cuyos sistemas eléctricos trabajan a frecuencias de 50 Hz, se podría modificar el sketch final para extender el equipo para supervisar instalaciones cuyos sistemas eléctricos trabajan a 60 Hz.

El diseño del prototipo está pensado para trabajar en instalaciones domésticas, sin embargo, podría ser interesante introducir el equipo en instalaciones industriales donde puedan tener muchos equipos que consumen energía eléctrica y se podría supervisar desde una pantalla los equipos que están encendidos, apagados y mostrar cuanto están consumiendo.

Gracias al control que puede ejercer sobre los equipos sería capaz de encenderlos o apagarlos a una determinada hora o programar una alarma en el caso de que un equipo no debiera estar encendido a determinada hora.

4.2 Detección del estado de equipos que consumen energía eléctrica.

Para realizar la detección del estado (encendido o apagado) de los equipos que consumen energía se va a dividir en dos ensayos. El primero, se va a simular mediante una tensión continua de 5V y 0V, que se extraerán de los pines digitales 5V y GND respectivamente. El segundo ensayo se realizará con la detección de un equipo que estará formado por tres bombillas que se alimentarán directamente de un enchufe.

4.2.1. Detección del estado de equipos mediante simulación con pines digitales 5V y GND.

En primer lugar, se ha decidido realizar una simulación usando pines digitales para verificar el funcionamiento del sketch realizado en el Arduino Ide.

```

boolean a; //Declaración variable booleana a
void setup() {
  Serial.begin(115200); //Establecimiento velocidad de comunicación
  pinMode(5, INPUT); //Configuración del pin 5 como entrada.
}

void loop() {
  a=digitalRead(5); //lectura del pin digital 5
  if(a==1){ //recibe un 1=5V
    Serial.println ("el equipo esta encendido");
  }
  else { //Recibe un 0=0V
    Serial.println("el equipo esta apagado");
  }
  delay(3000); //tiempo de espera para facilitar la lectura
}

```

el equipo esta encendido
el equipo esta apagado
el equipo esta encendido
el equipo esta apagado

Figura 11. Instrucciones ensayo de detección de estado con pines digitales y resultados

Para simular un equipo encendido, se ha decidido extraer una tensión una tensión continua de 5V, utilizando el pin digital de la misma placa 5V, conectando con un cable al pin digital 5 desde donde se realiza la lectura. Como muestra la figura 9, al utilizar la función *digitalRead(5)* (lee el estado del pin digital 5) y estar conectado a una tensión continua de 5V, esta función leerá un 1 binario (HIGH), lo que simulará que el equipo está encendido.

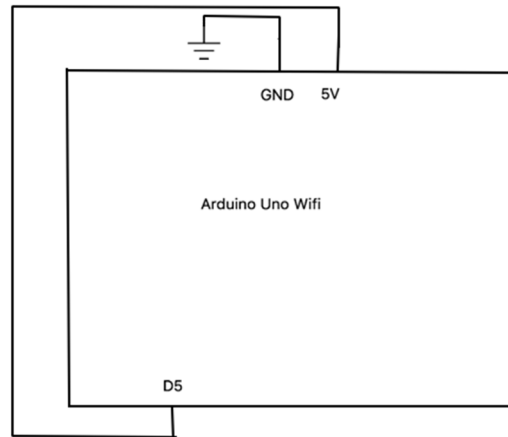


Figura 12. Esquema de montaje equipo encendido

Por otro lado, para simular un equipo apagado se ha conectado mediante un cable el pin digital 5 al pin digital GND, al utilizar la función *digitalRead (5)*, esta función leerá un 0 binario (LOW), lo que simulará que el equipo está apagado.

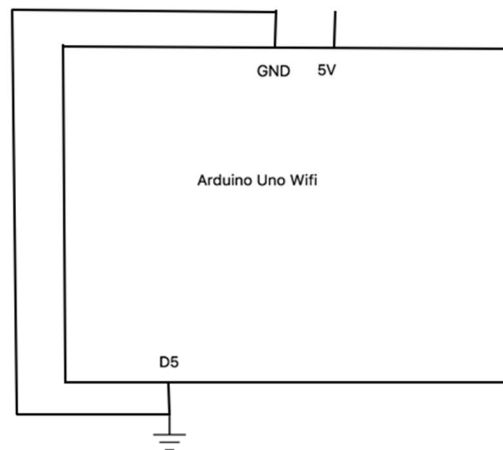


Figura 13. Esquema de montaje equipo apagado.

4.2.2. Detección del estado de equipos mediante puente de diodos y optoacoplador.

En este apartado se va a explicar cómo se realizará la detección de un equipo que está consumiendo energía eléctrica, para ello necesitaremos 3 resistencias, una de $330\text{K}\Omega$, otra de $6,8\text{K}\Omega$ y una última de $10\text{K}\Omega$, 1 condensador de 100 nF y otro de $10\text{ }\mu\text{F}$, un puente de diodos, un optoacoplador 4N25 y un diodo.

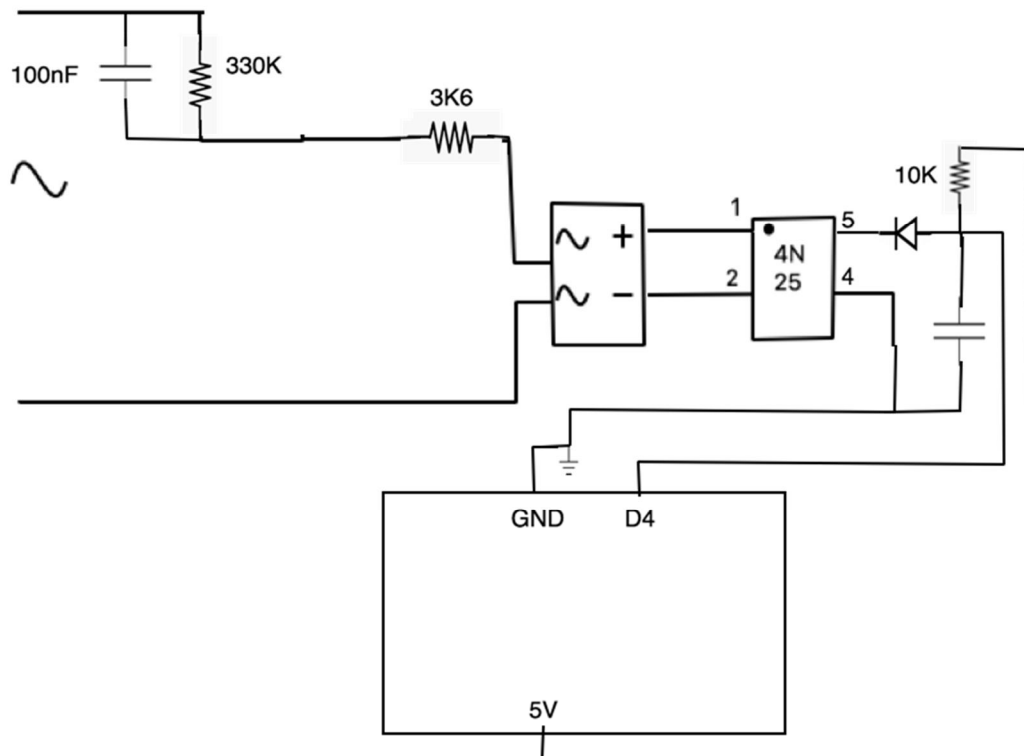


Figura 14. Esquema de montaje Detección de estado.

Una vez montado el esquema, que se puede observar en la figura 14, se conecta mediante un cable el cátodo del diodo con el pin digital 5 para realizar la lectura digital del equipo que se va a comprobar su estado. El montaje está diseñado de manera que cuando el equipo esta encendido en el punto X habrá un 0 binario (LOW), en el caso contrario, cuando el equipo esté apagado en el punto X habrá un 1 binario (HIGH), por lo que habrá que modificar el sketch anterior para realizar la medida.

```

boolean a; //Declaración variable booleana a
void setup() {
  Serial.begin(115200); //Establecimiento velocidad de comunicación
  pinMode(5, INPUT); //Configuración del pin 5 como entrada.
}

void loop() {
  a=digitalRead(5); //lectura del pin digital 5
  if(a==0){ //recibe un 0=0V
    Serial.println ("el equipo esta encendido");
  }
  else { //Recibe un 1=5V
    Serial.println("el equipo esta apagado");
  }
  delay(3000); //tiempo de espera para facilitar la lectura
}

```

```

el equipo esta encendido
el equipo esta apagado
el equipo esta encendido
el equipo esta apagado

```

Figura 15. Instrucciones ensayo de detección de estado con optoacoplador y resultados.

4.3 Control del estado de equipos.

La función de controlar el estado del equipo se basa en poder encender y apagar los distintos equipos que pueda tener la instalación a comprobar. Para conseguir controlar el encendido y apagado de los equipos se ha decidido realizar dos ensayos, el primero utilizando solo la placa Arduino Uno Wifi, aprovechando el led que tiene en el pin 13. Y un segundo utilizando un montaje con varias bombillas y el circuito integrado *MOC 3040*.

4.3.1 Control mediante la placa Arduino Uno Wifi únicamente.

La finalidad de este ensayo es familiarizarse con la programación de los sketches, para ello aprovechando que la placa en el pin 13 posee un led, crearemos un sketch el cual configurará el pin 13 como una salida y en función del valor binario que mandemos por el monitor serie se encenderá o se apagará la luz led que simulará el equipo que se quiere controlar, de manera que si enviamos un 1 binario (HIGH) la luz led se encenderá y simulará que el equipo se ha encendido. Por otro lado, si enviamos un 0 binario (LOW) por el monitor serie la luz se apagará simulando que se ha apagado el equipo.

```
const int pin = 13;
int option;

void setup(){
  Serial.begin(115200);
  pinMode(pin, OUTPUT);
}

void loop(){
  //si existe información pendiente
  if (Serial.available()>0){
    //leemos la opción
    char option = Serial.read();
    if (option == '0' ) //si el valor es 0
    {
      digitalWrite(pin, LOW); //apagamos el pin
    }
    if (option == '1' )
    {
      digitalWrite(pin, HIGH); //encendemos el pin
    }
    delay(200);
  }
}
```

Figura 16. Instrucciones sketch de control pin 13.

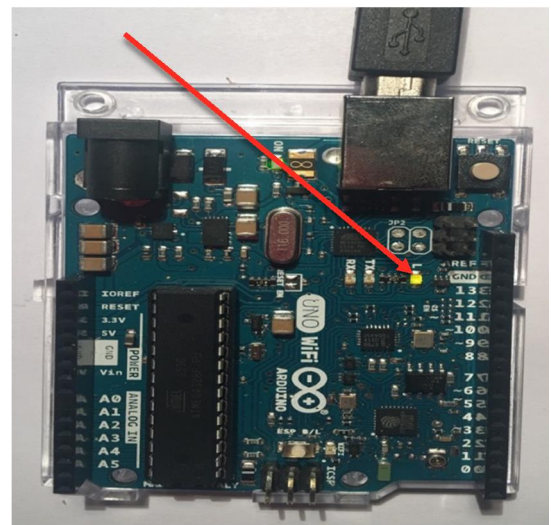
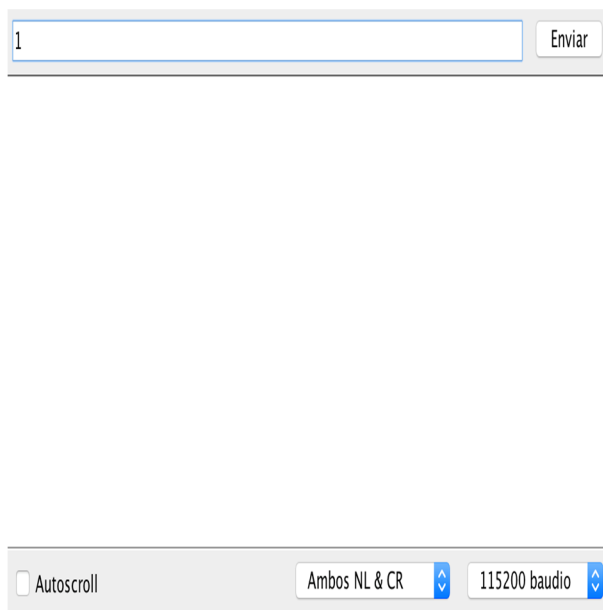


Figura 17. Instrucción de control encenderse y resultado en placa.

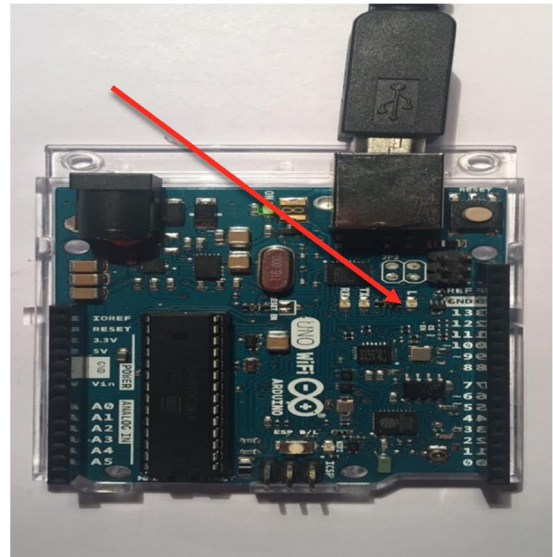
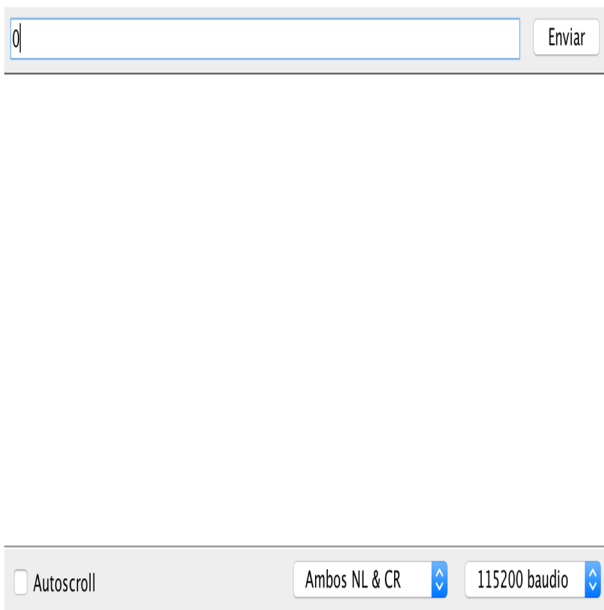


Figura 18. Instrucción de control apagar y resultado en placa.

4.3.2. Control mediante la placa Arduino Uno Wifi y optoacoplador fototriac.

En este ensayo se va a simular el encendido y apagado de un equipo, este equipo consistirá en una caja con tres bombillas, en el lateral de la caja hay seis conectores para poder configurar el conjunto de bombillas en función de la necesidad (1, 2 o 3 bombillas conectadas en serie y/o paralelo).

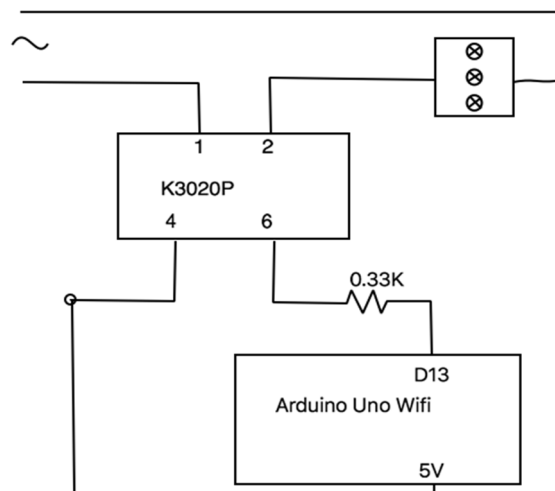


Figura 19. Esquema de montaje ensayo control con optoacoplador fototriac.

Para realizar el montaje del esquema se va a necesitar una resistencia de $0.38\text{ K}\Omega$, para ajustar la intensidad a 10 mA .

Se configurará el pin digital 13 como salida. Se ha elegido este pin para realizar el montaje debido a que al estar acompañado de una luz led en la placa y al configurarse como salida, está luz se encenderá en el caso de que se envíe un 1 binario (HIGH) y se apagará si se envía un 0 binario (LOW), servirá de ayuda visual para saber que se está enviando al pin digital y comprobar el correcto funcionamiento del montaje.

El optoacoplador fototriac K3020P funciona de manera que si entre las patas 4 y 6 circula corriente en este caso de 10 mA , permitirá la circulación de corriente entre las patas 1 y 2, por el contrario si por las patas 4 y 6 no circula corriente, cortará la corriente que circula por las patas 1 y 2.

Teniendo en cuenta la explicación anterior del funcionamiento del optoacoplador fototriac K3020P y el esquema de montaje de la figura 17, se podrán realizar las siguientes acciones:

- Encender las bombillas. Para realizar esta acción habrá que enviar al pin digital un 0 binario (LOW), lo que impondrá una tensión de 0V . Debido a esto la corriente (10mA) circulará desde la pata 4 a la pata 6, lo que provocará que la corriente procedente de la red eléctrica circule entre las patas 1 y 2 cerrando el circuito y encendiendo las bombillas. Para comprobar el correcto funcionamiento del montaje habría que comprobar que la luz led que acompaña al pin digital 13 está apagada.

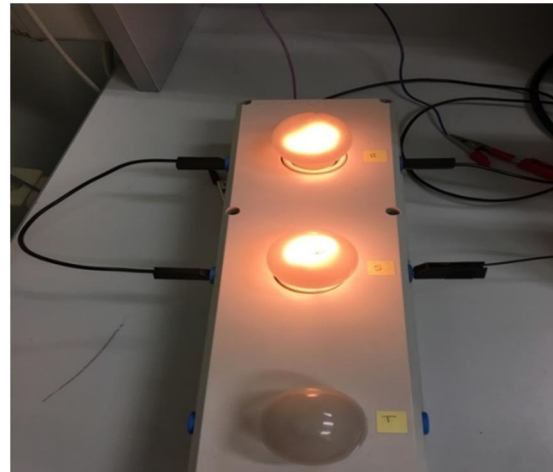
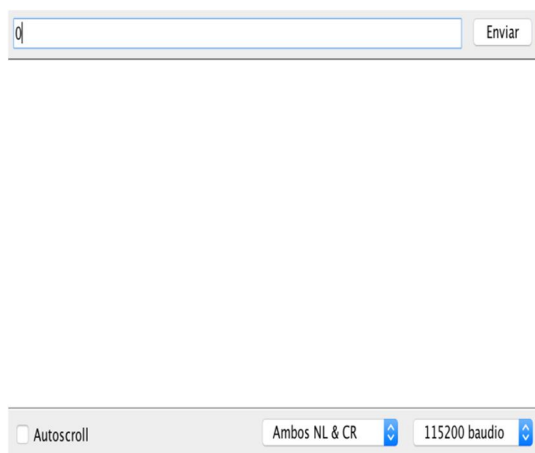


Figura 20. Envío de orden para encender las bombillas y resultado.

- Apagar las bombillas. Para apagar las bombillas, habrá que mandar un 1 binario (HIGH) al pin digital 13, que impondrá una tensión de 5V , debido a esto no será posible conducir corriente entre las patas 4 y 6 del optoacoplador fototriac K3020P, al no circular intensidad entre las patas 4 y 6 cortará la conducción de intensidad entre las patas 1 y 2 que provocará que se apaguen las bombillas.

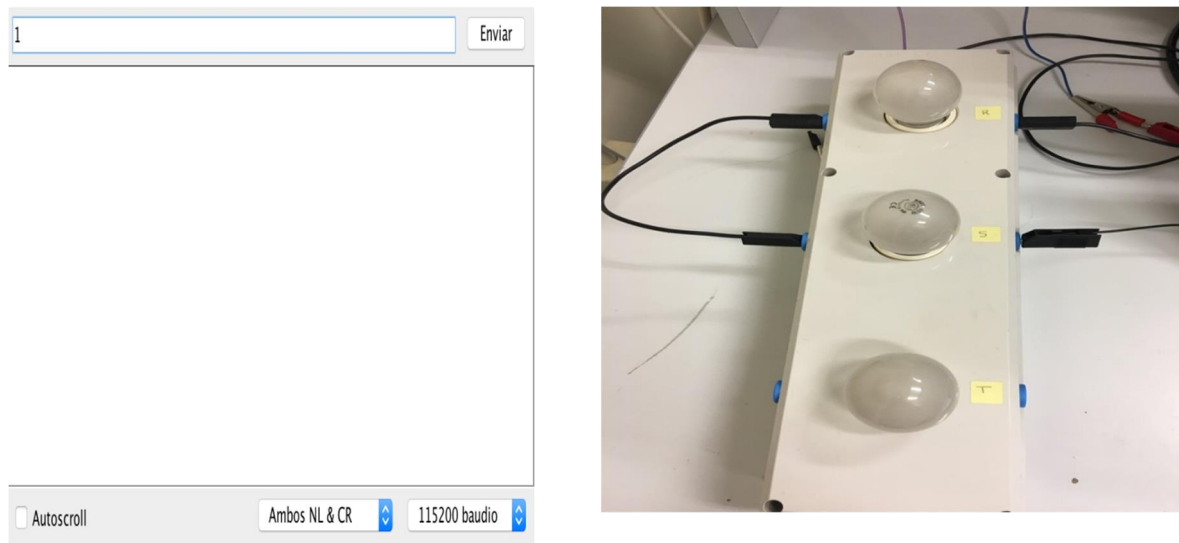


Figura 21. Envío orden para apagar las bombillas y resultado.

4.4 Medida de intensidad y cálculo de potencia aparente.

Para realizar la medida de intensidad se utilizará uno de los pines analógicos (A0), cada entrada analógica tiene 10 bits de resolución (1024 posibles valores) y un rango de 0-5 V. La entrada analógica utilizando la función *analogRead(A0)* leerá un valor de los 1024 valores posibles que se deberá convertir mediante unas operaciones matemáticas para pasar de bits a magnitudes reales.

Debido a que Arduino mide tensiones entre 0-5V para calcular la intensidad que consume el equipo a medir habrá que multiplicar la tensión leída por un factor K de conversión que se ajustará en los ensayos.

Se ha decidido calcular la potencia aparente que consume el equipo a medir suponiendo una tensión efectiva de 230V, una posible mejora del equipo de supervisión podría ser obtener la medida de tensión que consume el equipo a supervisar y a partir de los datos obtenidos tanto de intensidad como tensión calcular la potencia activa y reactiva, pudiendo calcular a partir de estos datos el factor de potencia.

Los pasos que se han seguido para realizar el ensayo han sido los siguientes:

- Medida de una tensión continua por la placa y un generador de ondas.
- Medida de una tensión alterna por la placa, osciloscopio y generador de ondas.
- Medida de intensidad

4.4.1 Medida de una tensión continua.

Para realizar la medida de tensión continua solo se necesitará un generador de ondas, un multímetro y la placa Arduino Uno Wifi.



Figura 22. Generador de ondas

Como se puede observar en la figura 22, para establecer una tensión continua habrá que marcar el botón *DC* y desmarcar tanto el botón de onda rectangular, triangular y senoidal, de manera que el tipo de onda que se generará será una onda continua. Aunque podemos generar ondas continuas desde -5V a +5V, para realizar el ensayo generaremos ondas desde 0V a +5V debido a que las tensiones negativas pueden perjudicar gravemente las entradas analógicas de la placa.

```
void loop() {

    // lee el pin de entrada analogico (pin A0)

    int valor_sensor = analogRead(A0);
    // Convierte la lectura analogica (0 - 1024) hacia un voltaje (0-5V):
    voltaje = (valor_sensor) * (5.0 / 1024.0);

    Serial.println(voltaje);
    delay(3000);
}
```

Figura 23. Instrucciones sketch medida

```
0.00
0.00
0.00
0.00
0.00
0.00
0.55
0.80
1.07
1.75
2.48
3.24
4.53
5.00
5.00
5.00
5.00
tensión continua y resultados.
```


Para realizar el ensayo se ha programado un sketch que lee la tensión de entrada en la entrada analógica A0 cada tres segundos y lo imprime por pantalla. El valor se lee en bits por lo que será necesario convertirlo a un valor real utilizando la fórmula (1), donde Vr es el valor real y Vb es el valor leído en bites.

$$Vr = Vb * \frac{5}{1024} \quad (1)$$

Para comprobar la eficiencia de la medida se ha conectado un multímetro para conocer la magnitud real de la tensión suministrada. Obteniéndose los siguientes resultados:

Medida multímetro (voltios)	Medida Arduino (voltios)	Error absoluto (%)
0.26	0.26	3.85
1.09	1.13	3.66
2.02	2.09	3.47
2.05	2.13	3.90
3.17	3.31	4.42
4.08	4.25	4.17
4.83	5.0	3.52

Tabla 2. Medidas de tensión continua.

Calculándose el error absoluto utilizando la fórmula (2). Donde Vm es la medida de tensión obtenida por el multímetro y Va es la medida de tensión obtenida por la placa.

$$E(\%) = \frac{|Vm - Va|}{Va} * 100 \quad (2)$$

Aunque la medida con la placa tiene un error absoluto muy pequeño (menor del 5%), se va aplicar un factor de corrección para reducir este error. Para calcular el factor de corrección se eleva la tensión hasta el máximo valor que puede leer sin salir del rango (el máximo valor antes de que la medida en la placa sea 5.0). Obteniéndose las siguientes medidas:

Medida multímetro (voltios)	Medida Arduino (voltios)
4.63	4.83

Tabla 3. Medidas de tensión para calcular el factor de corrección.

Para calcular el factor de corrección K se divide la tensión medida por el multímetro por la tensión obtenida por la placa.

$$K = \frac{4,63}{4,83}$$

$$K = 0.96$$

A continuación, se multiplica cada medida de tensión obtenida por la placa por este valor de corrección obteniéndose las siguientes medidas corregidas y el nuevo valor absoluto corregido:

Medida multímetro (voltios)	Medida Arduino (voltios)	Error absoluto (%)	Medida Arduino corregida (voltios)	Error absoluto corregido (%)
0.26	0.26	3.85	0.258	0.45
1.09	1.13	3.66	1.08	0.62
2.02	2.09	3.47	2.00	0.81
2.05	2.13	3.90	2.04	0.4
3.17	3.31	4.42	3.17	0
4.08	4.25	4.17	4.07	0.15
4.83	5.0	3.52	4.79	0.77

Tabla 4. Medidas de tensión continua aplicando factor de corrección.

Como se puede comprobar con la tabla 3, se puede reducir mucho el error absoluto aplicando un factor de corrección, consiguiendo que la medida obtenida sea muy fiable ya que el error absoluto es menor del 1%.

4.4.2 Medida de una tensión alterna utilizando el generador de ondas y la placa Arduino.

La finalidad de este ensayo es comprobar que la placa es capaz de medir una tensión eficaz. Para ello se ha realizado una simulación de las medidas que se obtendrán en el siguiente ensayo (medida de intensidad por la placa arduino), se ha generado con el generador de ondas una tensión con forma senoidal centrada en 2.5 voltios, la forma de generar esta onda es aplicar una suma de dos tensiones, una tensión continua de 2.5 voltios y una tensión alterna senoidal, la cual variaremos su amplitud para ver el comportamiento del Arduino al medir.

Para calcular el valor eficaz de la tensión con la placa Arduino, se han realizado dos ensayos. En el primero, se mide una onda completa. Para ello se obtienen 40 puntos de tensión instantánea, separados 0.5 milisegundos y aplicando la fórmula (3), donde V_{ef} es el valor de la tensión eficaz, V_i es el valor de la tensión instantánea y n es el número de medidas. En el segundo ensayo, se miden dos ondas completas, para ello se obtienen 80 puntos de tensión instantánea separados 0.5 milisegundos.

$$V_{ef} = \frac{\sqrt{\sum V_i^2}}{n} \quad (3)$$

En primer lugar, se ha generado una onda senoidal de amplitud 2.5 voltios, hay que tener en cuenta que a la hora de programar el sketch de medida hay que restar la tensión de referencia.

Mediante un multímetro se ha obtenido el valor eficaz real que se ha generado, dando una tensión eficaz de 1.7 voltios. Mientras que en la placa Arduino se han obtenido valores cercanos a 1.7 V, se puede observar que tiene cierta dispersión como se puede comprobar en la tabla 4.

Número de medida	Tensión eficaz medida una onda (voltios)	Tensión eficaz medida dos ondas (voltios)
1	1.70	1.73
2	1.80	1.74
3	1.84	1.63
4	1.73	1.79

Tabla 5. Medidas de tensión eficaz I.

En segundo lugar, se ha generado una onda senoidal de amplitud 0.707 voltios. Mediante un multímetro se ha medido la tensión eficaz que se ha generado, siendo de 0.5 voltios. Por otra parte, se ha medido utilizando la placa, obteniendo valores cercanos a 0.5 voltios, también hay cierta dispersión de datos como se puede ver en la tabla 5.

Número de medida	Tensión eficaz medida (voltios)
1	0.50
2	0.52
3	0.55
4	0.48

Tabla 6. Medidas de tensión eficaz II.

Por último, se ha generado una onda senoidal de amplitud 0.5656 voltios. Con la ayuda de un multímetro se ha medido la tensión eficaz real, resultando de 0.4 voltios. Como en los casos anteriores, se ha medido la tensión eficaz utilizando la placa, obteniendo valores cercanos a 0.4 voltios, aunque como en los casos anteriores también hay una dispersión en los valores, como se puede apreciar en la tabla 6.

Número de medida	Tensión eficaz medida (voltios)
1	0.42
2	0.41
3	0.40
4	0.38

Tabla 7. Medidas de tensión eficaz III.

En función de los anteriores resultados se pueden obtener las siguientes conclusiones:

- Se puede apreciar que el error de la medida aumenta proporcionalmente con el aumento de la tensión a medir.
- Como se puede comprobar en la tabla 4, la dispersión de valores es mayor en la medida de una onda que en la de dos ondas. Se puede afirmar que cuanto mayor sea el número de valores de tensión instantánea, menor es el error cometido. Es por esto que para los siguientes ensayos se han medido directamente dos ondas completas.

4.4.3 Medida de intensidad.

Para medir la intensidad que circula vamos a realizar el montaje de la figura 24. Para ello debemos tener en cuenta las siguientes consideraciones:

- Se utiliza un transformador aislador de relación 1:1 entre sus devanados primario y secundario para aislarnos de la red.
- Se va a acoplar el circuito integrado TL 341, como se ha explicado anteriormente se utiliza para referenciar la tensión a 2.5 voltios, de manera que centrará la tensión de medida en este valor para poder medir la tensión por el Arduino.

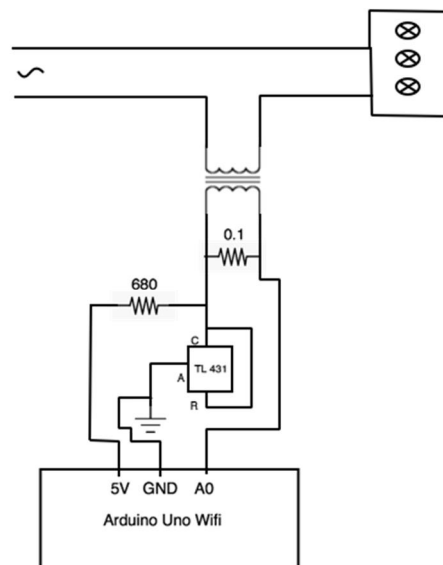


Figura 24. Medida de Intensidad

Por otra parte, se tiene que tener en cuenta que las entradas analógicas de la placa pueden medir tensión desde 0 a 5 voltios, es por esto que se deberá ajustar la medida multiplicando por un factor para realizar la medida de la intensidad.

Este ensayo se ha dividido en las siguientes etapas:

- Cálculo de la constante para transformar tensión en intensidad.
- Medida de intensidad de tres bombillas en paralelo.
- Medida de intensidad de dos bombillas en paralelo.
- Medida de intensidad de una bombilla.

4.4.3.1. Cálculo de la constante para transformar tensión en intensidad.

Para realizar este cálculo se han configurado las tres bombillas en paralelo (para obtener la mayor intensidad posible), se ha conectado a la caja de las bombillas en serie un multímetro (en la posición amperímetro), para medir la intensidad real. Y mediante la placa se ha medido la tensión eficaz que será proporcional a la intensidad que se quiere medir, obteniéndose los datos de la tabla 7.

Intensidad medida por el multímetro (amperios)	Tensión eficaz medida por la placa (voltios)
0,516	0.1

Tabla 8. Medidas intensidad real (multímetro) y tensión eficaz (amperímetro).

Utilizando la fórmula (4) obtenemos el factor de corrección Q, donde I es la intensidad real medida por el multímetro y V es la tensión eficaz medida por la placa. Obteniendo un valor $Q=5.15 \text{ A/V}$

$$Q = \frac{I}{V} \text{ (A/V)} \quad (4)$$

4.4.3.2. Medida de intensidad de tres bombillas en paralelo y cálculo de la potencia aparente.

Para el siguiente ensayo se realiza el montaje de la figura 25, configurando las tres bombillas en paralelo. Se medirá la intensidad real que recorre la caja de las bombillas con un multímetro conectado en serie con la caja de bombillas. A continuación, se muestra un reportaje fotográfico del ensayo.

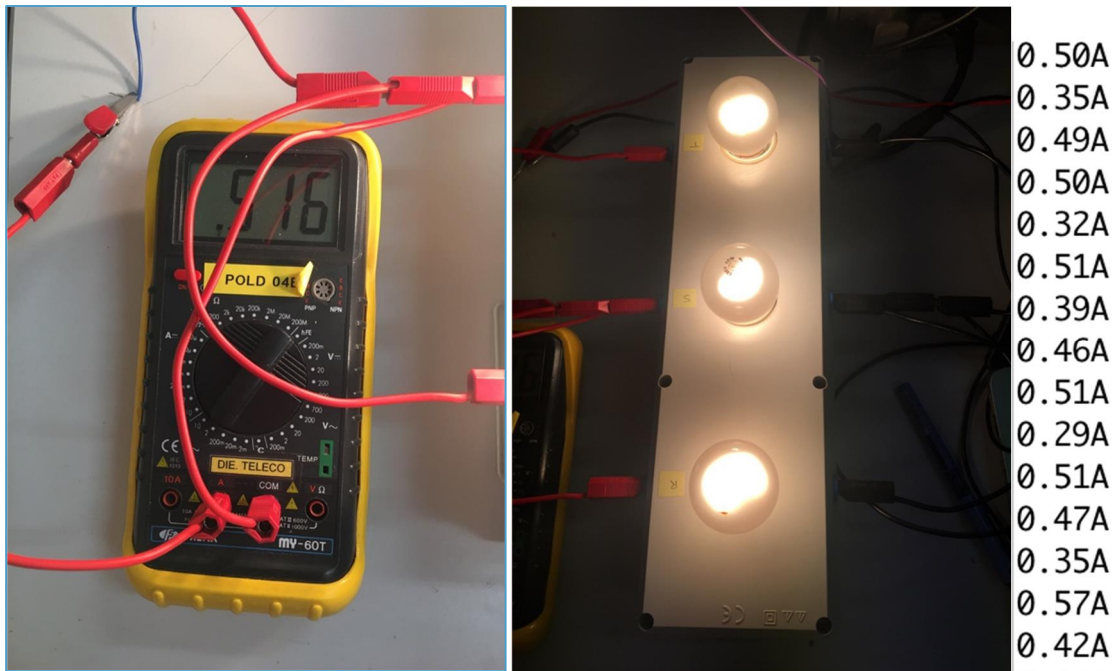


Figura 25. Imágenes de intensidad real, estado de las bombillas y medidas de la intensidad por la placa (I).

Tras el ensayo se observa que la medida obtenida por la placa tiene oscilaciones como también se puede observar en la figura 25. Este efecto se debe a que no se puede elegir el momento en el que empieza medir las ondas. Este efecto se puede reducir aumentando la cantidad de las medidas para calcular la intensidad eficaz, para esto bastaría con reducir el tiempo de espera entre medida y medida o manteniendo el tiempo de espera entre medidas y aumentando el número de medidas midiendo más de dos ondas completas.

Para calcular la potencia aparente se supone una tensión eficaz de 230V y aplicando la fórmula (5), donde V es la tensión eficaz de 230V e I es la intensidad calculada por la placa.

$$P = V * I \text{ (W)} \quad (5)$$

La intensidad es de 0.50A
 La tensión es de 230V
 La potencia aparente es de 115.00VA
 La intensidad es de 0.52A
 La tensión es de 230V
 La potencia aparente es de 119.60VA
 La intensidad es de 0.47A
 La tensión es de 230V
 La potencia aparente es de 108.10VA

Figura 26. Cálculo potencia aparente a partir de la intensidad medida por la placa (I).

4.4.3.3. Medida de intensidad de dos bombillas en paralelo y cálculo de la potencia aparente.

Este ensayo es similar al anterior, al tener una bombilla menos la intensidad que circulará será menor en comparación con el ensayo anterior. Se va a realizar el mismo procedimiento, calcular la intensidad real con un multímetro y medir con la placa la intensidad que circula. A continuación, se muestra un reportaje fotográfico del ensayo.

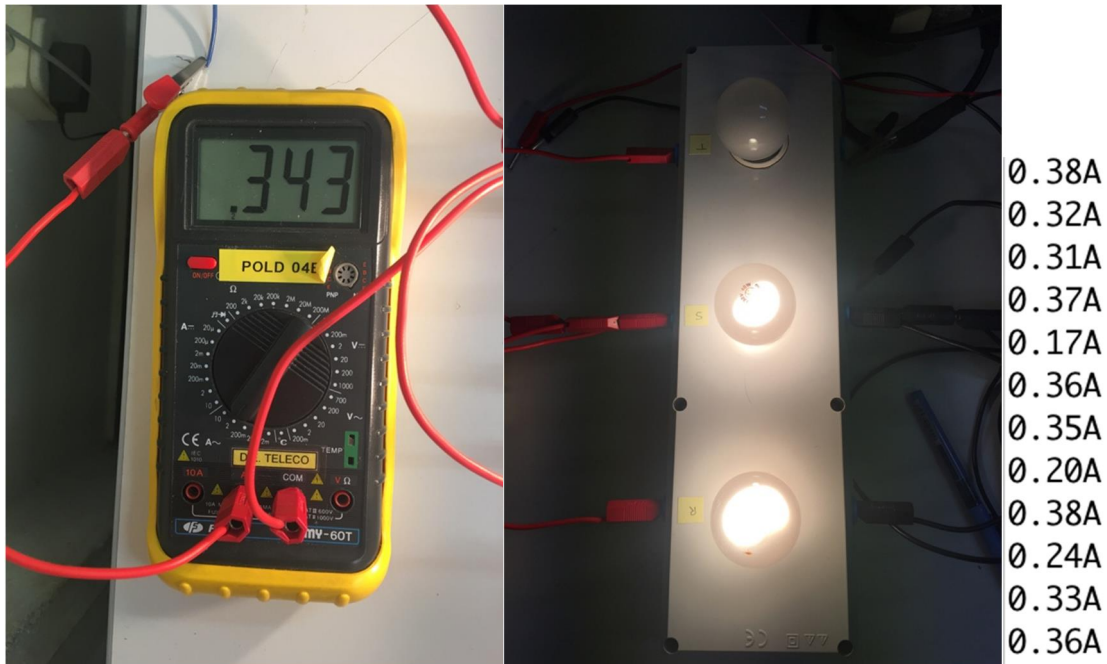


Figura 27. Imagen de intensidad real, estado de las bombillas y medidas de la intensidad por la placa (II).

Por último, en la figura 27, se muestra el resultado del cálculo de la potencia aparente por la placa.

La intensidad es de 0.38A
 La tensión es de 230V
 La potencia aparente es de 87.40VA
 La intensidad es de 0.33A
 La tensión es de 230V
 La potencia aparente es de 75.90VA
 La intensidad es de 0.35A
 La tensión es de 230V
 La potencia aparente es de 80.50VA

Figura 28. Cálculo potencia aparente a partir de la intensidad medida por la placa (II).

4.4.3.4. Medida de intensidad de una bombilla y cálculo de la potencia aparente.

Para este ensayo solo tenemos una bombilla conectada, es por esto que la intensidad se vuelve a reducir con respecto a los dos ensayos anteriores.

Actuando de manera análoga a los ensayos anteriores, se mide la intensidad real que circula por la bombilla con un multímetro y se calcula la intensidad efectiva con la placa. A continuación, se muestra un reportaje fotográfico del ensayo.



Figura 29. Imágenes de intensidad real, estado de las bombillas y medidas de la intensidad por la placa (III).

A continuación, como en los dos ensayos anteriores, se muestra la figura 30 donde se puede apreciar el cálculo de la potencia aparente.

La intensidad es de 0.19A
La tensión es de 230V
La potencia aparente es de 43.70VA
La intensidad es de 0.12A
La tensión es de 230V
La potencia aparente es de 27.60VA
La intensidad es de 0.17A
La tensión es de 230V
La potencia aparente es de 39.10VA

Figura 30. Cálculo potencia aparente a partir de la intensidad medida por la placa (III).

4.4.3.5. Conclusiones de la medida de la intensidad y cálculo de la potencia aparente.

A partir de los resultados obtenidos en los ensayos se obtienen las siguientes conclusiones:

- La dificultad de no poder decidir en qué momento se desea empezar a medir la onda puede provocar las oscilaciones que se aprecian al observar las medidas de la intensidad por la placa.
- Se observa como las oscilaciones aumentan proporcionalmente con el aumento de la intensidad que recorre el equipo.
- La idea de calcular la potencia aparente que consume el equipo a medir se basa en dar un orden de magnitud del consumo.

Capítulo 5. Diseño final.

En el capítulo 5, se va a explicar el funcionamiento del equipo en conjunto, como será la alimentación y el montaje final.

5.1. Funcionamiento del equipo

Una vez realizados los ensayos anteriores, se procede a realizar el diseño final. Para ello, se ha decidido dividir el programa en dos sketches distintos, los dos sketches se pueden consultar en el anexo 2, al igual que el esquema final de montaje que se encuentra en el anexo 1.

En primer lugar, se ha desarrollado una interfaz gráfica para controlar el equipo mediante el software *Processing*, el cual gracias a la comunicación serie recibirá de la placa el estado de los electrodomésticos que se van a controlar y transmitirá las ordenes de encendido y apagado a la placa.

La interfaz gráfica consta de dos partes diferenciadas:

- La primera es un plano de una instalación muy sencilla, una habitación con cuatro electrodomésticos cada uno de ellos ubicado en una esquina de la habitación, se va a simular tres de ellos asociándolos a los pines digitales 3,5 y 6, de manera que si se conecta una tensión continua de 5 voltios a cada uno de estos pines digitales, la placa leerá un 1 binario (HIGH), esto simularía que el electrodoméstico está encendido y por lo tanto se encendería una luz verde en el plano de la interfaz. Si por el contrario, se conecta una tensión de 0 voltios el funcionamiento sería el contrario, estos pines digitales leerían un 0 binario (LOW), lo que simularía que están apagados y en el plano se encendería una luz roja en la posición de cada electrodoméstico que este apagado. El pin digital 4 se reserva para la lectura mediante el circuito integrado 4N25. Como se ha demostrado en el ensayo, en el anterior capítulo, el funcionamiento será el contrario. Si simulamos que el electrodoméstico está encendido, en el cátodo del diodo, que se conecta al pin digital 4, se leerá un 0 binario (LOW), por lo que la luz que aparecerá será verde. Si simulamos que el electrodoméstico está apagado, en el cátodo del diodo se leerá un 1 binario (HIGH), debido a esto la luz será de color rojo.
- La segunda parte de la interfaz es un botón, este botón será el encargado de controlar el estado de la iluminación de la habitación, la luz se colocará en el centro de la habitación. Si la luz se enciende, aparecerá un mensaje que indica "LUZ ON" y se indicará una luz verde en la localización de la luz en el plano. sin embargo, si la luz se apaga el mensaje que saldrá es "LUZ OFF" mientras que la luz en el plano sería de color rojo.



Figura 31. Interfaz gráfica (I).



Figura 32. Interfaz gráfica (II).

En la figura 31 y figura 32 Se puede observar como el electrodoméstico de la esquina superior derecha (asociado al pin digital 4) funciona al revés que los otros electrodomésticos.



Figura 33. Interfaz gráfica (III).



Figura 34. Luz encendida.

Como se puede apreciar en la figura 33 se ha encendido la luz en la interfaz gráfica, acompañada de la luz física del montaje en el laboratorio, figura 34. Se puede ver tanto el mensaje de “LUZ ON” como la luz de color verde del plano en la interfaz gráfica.



Figura 35. Interfaz gráfica (IV).

A continuación, se pulsa el botón para apagar la luz, como aparece en la figura 35, tanto el color de la luz que indica el plano como el mensaje del botón han cambiado. Y como se comprueba en la figura 36, la luz física del montaje también se apaga.

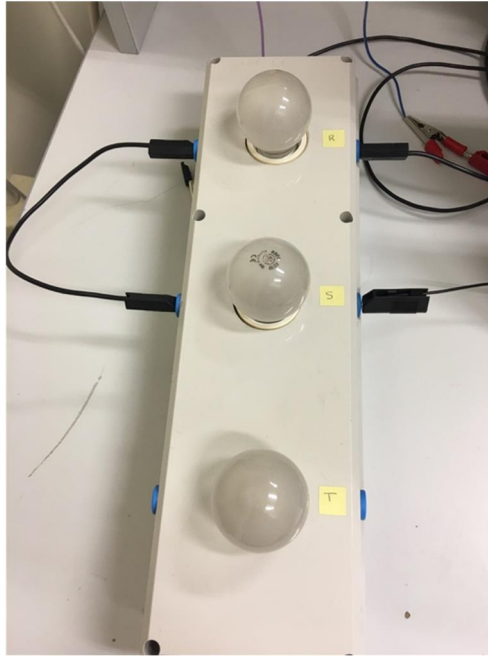


Figura 36. Luz apagada.

En segundo lugar, el sketch que mide la intensidad y el cálculo de la potencia aparente que consume el electrodoméstico será el mismo que se ha utilizado durante el ensayo. Debido a que se ha podido reducir las oscilaciones, pero no eliminar, se ha decidido que, para que el usuario tenga un orden de magnitud de la potencia aparente que consume el electrodoméstico, la placa se comunicará con el monitor serie informando de la intensidad medida y la potencia aparente.

5.2. Alimentación del equipo.

El equipo actualmente está diseñado para que se conecte mediante cable USB a un PC y que este le suministre la tensión necesaria. No obstante, la placa permite la conexión mediante el puerto Wifi, en una futura actualización en la que se utilice la conexión Wifi para comunicarse con un PC o cualquier otro dispositivo que lo vaya a controlar necesitaría el diseño de una fuente de alimentación. Se ha pensado en dos opciones:

Las dos posibles fuentes de alimentación son:

- Pila de 9V o batería recargable.
- Alimentación desde la red: implicaría el uso de un transformador para reducir la tensión, un puente rectificador de diodos para convertir tensión alterna en tensión continua y finalmente se introduciría un condensador para disminuir el rizado.

5.3. Montaje.

El equipo finalmente irá montado en una caja ya que aporta mayor seguridad y para tener un mejor acabado, ya que la función es supervisar una instalación eléctrica doméstica y la estética de la vivienda es un factor que preocupa mucho a los usuarios. La caja deberá instalarse cerca de la

ubicación de un PC o en un lugar en el que se pueda colocar un ordenador debido a que este va conectado al equipo mediante un cable USB, que saldría de dentro de la caja.

Como se ha explicado antes, en el futuro caso de utilizar la conexión Wifi, habría que mantener el cable USB ya que tanto para la primera configuración de la red Wifi como para posteriores configuraciones es necesario conectar la placa al PC.



Figura 37. Caja Tempo de Fibox 187x122x90 cm

Como se puede observar en la figura 37, sus dimensiones son de 187mm de ancho 122 de alto y 90 de largo. Se ha escogido esta caja porque es robusta, resistente a los impactos, es económica y tiene un rango de protección de temperaturas de -40°C a 60°C .

Sería conveniente realizar un circuito impreso y soldar los elementos a este, ya que aumentaría la fiabilidad del equipo frente a malas conexiones

Capítulo 6. Conclusiones.

En este capítulo se va a tratar acerca de las posibles mejoras, la problemática que ha surgido a la hora de realizar el proyecto y las conclusiones del mismo en función de si se han cumplido los objetivos marcados al inicio.

6.1. Posibles mejoras.

Optimización de las funciones de la placa Arduino Uno Wifi:

- Debido a que la placa admite hasta seis entradas analógicas se podría medir la tensión a la que se suministran los electrodomésticos que se desean supervisar, como consecuencia de medir tanto la intensidad como la tensión, se podría calcular el ángulo de desfase entre las ondas de tensión e intensidad, es decir el factor de potencia, o la medida de la potencia reactiva, ya que las compañías eléctricas penalizan el exceso de potencia reactiva.
- Al llevar incorporada la shield que permite conectarse a una red Wifi, se podría diseñar una APP que permitiera el control desde una Tablet o un Smartphone, de manera que pueda ser más atractivo y cómodo para el usuario.
- Incorporar un sistema de alarmas para que avise al usuario del momento en el que el precio de la luz está más caro o más barato, así mismo se podría programar para que encienda o apague los electrodomésticos a determinada hora o en función de señales que pueda recibir desde el exterior como pudiera ser la temperatura.

Extensión a otros tipos de instalaciones:

- El proyecto está destinado a la supervisión de instalaciones domésticas, aunque podría extenderse a las instalaciones industriales, ya que una instalación eléctrica industrial tiene muchos más equipos que consumen mucha más energía eléctrica que los electrodomésticos de una vivienda y podría ser interesante detectar mirando una pantalla los equipos que están encendidos o apagados de cara al cierre de la jornada laboral de esta industria, ya que si se olvida apagar una máquina la empresa asume costes innecesarios .

6.2. Problemática encontrada al realizar el proyecto.

La placa Arduino Uno Wifi ha sido capaz de medir la intensidad, aunque la mayoría de las medidas se ajustaban dentro del rango permitido de error aceptable, había otras que se escapaban de este

rango, hay que destacar que utilizar el microcontrolador de Arduino no es la mejor opción para realizar medidas analógicas, ya que tiene una elevada falta de capacidad de precisión.

El error que se aprecia en las mediciones puede ser debido a alguna de estas causas o a una suma de ellas.

- Falta de precisión a la hora de iniciar la medida de la onda. No se puede saber en qué momento empieza a medir la onda, sin embargo, sí que se demuestra que la lectura de tensión continua es bastante fiable mientras que la lectura de tensión alterna no lo es tanto. Esto puede ser debido a que en algunas ocasiones la medida puede empezar en una cresta o en un valle o en cualquier otro punto intermedio.

- Diferente tiempo de muestras. Durante los ensayos, para establecer el tiempo de espera entre muestreos se ha podido comprobar que el tiempo de lectura de la señal analógica no es siempre el mismo, esto puede provocar que no se lea la onda completa o que se lea más de una o dos ondas completas.

- Falta de precisión. Al declarar variables tipo float (4 bytes cuyo rango es [-3,4028235+38, 3,4028235+38]) solo se muestran dos decimales.

Por último, la placa escogida es muy novedosa, y por ello pocos usuarios han trabajado con ella. Me gustaría destacar que se ha conseguido conectar a una red Wifi doméstica, pero ha sido imposible conectarla a cualquier red de la Universidad Politécnica de Valencia.

6.3. Conclusiones.

Como conclusión, se han cumplido los tres principales objetivos establecidos al principio del proyecto, estos eran:

- Detección del estado de electrodomésticos.
- Actuación sobre electrodomésticos.
- Medida de corriente consumida por electrodomésticos.

Se ha comprobado que el microcontrolador trabaja mejor con parámetros digitales que con parámetros analógicos, es por esto que el equipo es mucho más óptimo realizando los dos primeros objetivos que el tercero.

Por otro lado, no se ha encontrado en el mercado un equipo que sea capaz de realizar las anteriores funciones, debido a esto, no se ha podido realizar una comparación con el equipo que se ha desarrollado en este proyecto.

Desde el punto de vista electrónico, se ha aprendido mucho, sobre todo a la hora de trabajar tanto con el software Arduino Ide como con el software de Processing, aunque durante el Grado, en asignaturas de informática, se ha aprendido a programar en lenguaje C++, para trabajar con estos softwares se ha tenido que ampliar el conocimiento de programación en lenguaje Java.

Trabajando con la placa Arduino, se ha podido comprobar la capacidad que tiene para realizar cualquier tipo de proyecto a un precio bastante reducido, tal y como dijo su cofundador Massimo Banzi (2011). Cabe destacar la comunidad de makers que se ha formado alrededor de Arduino, totalmente libre y en la que se comparte todos los códigos y en caso de necesidad se presta a la colaboración entre usuarios para solventar cualquier dificultad. Me gustaría destacar que la placa se conecta sin ningún problema a redes Wifi en las que solo hay que poner la contraseña de la red. Sin

embargo, si hay que conectarse a una red en la que se pide un nombre de usuario y contraseña para autorizar la conexión del dispositivo a la red no es capaz de conectarse.

Por último, los ensayos se han realizado gracias al laboratorio de electricidad, donde se hallan los equipos de altas prestaciones, sin los cuales no se habría podido ensayar de manera tan óptima.

7. Catálogos.

- Catálogo circuito integrado referencia de tensión TL431 (<http://www.ti.com/lit/ds/symlink/tl431.pdf>)
- Catálogo circuito integrado 4N25 (<https://www.vishay.com/docs/83725/4n25.pdf>)
- Catálogo circuito integrado fototriac K3020P (<http://www.alldatasheet.com/datasheet-pdf/pdf/219574/VISHAY/K3020P.html>)
- Catálogo circuito integrado W08M (<http://www.futurlec.com/Diodes/W08M.shtml>)
- Catálogo caja Fibox Tempo (<http://www.media.automation24.com/datasheet/5814025.pdf>)

8. Bibliografía

Libros:

1. José R. Lajara Vizcaíno y José Pelegrí Sebastián (2014) Sistemas Integrados con Arduino. Marcombo
2. Oscar Torrente Artero (2015). El mundo Genuini-Arduino. Curso práctico de formación. ANAYA
3. Massimo Banzi (2011). Getting started with Arduino. Maker Media.

Páginas Web:

1. (2017) Convenio Colectivo del sector de empresas de ingeniería y oficinas de estudios técnicos, resolución de 30 de diciembre de 2016 de la Dirección General de Empleo y publicada en el BOE el 18 de enero de 2017.
Página Web: <https://www.boe.es/boe/dias/2017/01/18/pdfs/BOE-A-2017-542.pdf>
2. (2016) Configurar la placa Arduino Uno Wifi:
<https://aprendiendoarduino.wordpress.com/2016/11/12/wifi-en-arduino/>
3. (2016) Start with Arduino One Wifi:
<http://www.arduino.org/learning/getting-started/getting-started-with-arduino-uno-wifi/>
4. Sitio web Processing <https://www.processing.org/>

9. Resumen de Figuras

- Figura 1. Pantalla programa Arduino Ide.
- Figura 2. Pantalla del entorno de programación de Processing.
- Figura 3. Arduino Uno Wifi. Recuperado de <http://www.arduino.cc>
- Figura 4. Imagen y esquema del circuito integrado K3020P. Recuperado del catálogo Vishay.
- Figura 5. Imagen y esquema circuito integrado 4N25. Recuperado del catálogo de Vishay
- Figura 6. Imagen TI341y diagrama de bloque funcional. Recuperado del catálogo ON semiconductor.
- Figura 7. Imagen y esquema puente rectificador de diodos. Recuperado del catálogo de Futurlec
- Figura 8. Protoboard. Recuperado de <http://www.cetronic.es>
- Figura 9. Osciloscopio y Generador de ondas
- Figura 10. Multímetro
- Figura 11. Instrucciones ensayo de detección de estado con pines digitales y resultados
- Figura 12. Esquema de montaje equipo encendido
- Figura 13. Esquema de montaje equipo apagado.
- Figura 14. Esquema de montaje Detección de estado.
- Figura 15. Instrucciones ensayo de detección de estado con optoacoplador y resultados.
- Figura 16. Instrucciones sketch de control pin 13.
- Figura 17. Instrucción de control encenderse y resultado en placa.
- Figura 18. Instrucción de control apagar y resultado en placa.
- Figura 19. Esquema de montaje ensayo control con optoacoplador fototriac.
- Figura 20. Envío de orden para encender las bombillas y resultado.
- Figura 21. Envío orden para apagar las bombillas y resultado
- Figura 22. Generador de ondas
- Figura 23. Instrucciones sketch medida tensión continua y resultados.
- Figura 24. Medida de Intensidad
- Figura 25. Imágenes de intensidad real, estado de las bombillas y medidas de la intensidad por la placa (I).
- Figura 26. Cálculo potencia aparente a partir de la intensidad medida por la placa (I).
- Figura 27. Imagen de intensidad real, estado de las bombillas y medidas de la intensidad por la placa (II)
- Figura 28. Cálculo potencia aparente a partir de la intensidad medida por la placa (II).
- Figura 29. Imágenes de intensidad real, estado de las bombillas y medidas de la intensidad por la placa (III).
- Figura 30. Cálculo potencia aparente a partir de la intensidad medida por la placa (III).
- Figura 31. Interfaz gráfica (I).
- Figura 32. Interfaz gráfica (II).
- Figura 33. Interfaz gráfica (III).
- Figura 34. Luz encendida.
- Figura 35. Interfaz gráfica (IV).
- Figura 36. Luz apagada.
- Figura 37. Caja Tempo de Fibox 187x122x90 cm

PRESUPUESTO

1. Introducción.

El presupuesto se ha elaborado como consecuencia de un encargo externo para el diseño de un equipo de supervisión del funcionamiento de una instalación eléctrica basado en el uso de una placa Arduino.

2. Contenido del presupuesto.

Se ha consultado el Convenio Colectivo del sector de empresas de ingeniería y oficinas de estudios técnicos, resolución de 30 de diciembre de 2016 de la Dirección General de Empleo y publicada en el BOE el 18 de enero de 2017, para conocer los costes salariales de aplicación.

Se ha tomado como referencia el coste correspondiente al nivel 3: Técnico de cálculo o diseño, Jefe de 1ª y Programador de ordenador.

Descripción	Coste anual (€)	Coste (€/hora)
Salario base (225 días)	16917,60	9,40
Seguridad Social (28,3%)	4787,68	2,66
Dietas	0,00	0
Pluses	0,00	0
Total	21.705,28 €	12,06 €

Tabla 9. Desglose Del precio unitario de la mano de obra

Las horas empleadas en cada una de las partes del proyecto son:

Descripción	Cantidad (horas)	Precio unitario (€/hora)	Subtotal (€)
Planificación del equipo	40,00	12,06	482,34
Programación	40,00	12,06	482,34
Realización esquemas	70,00	12,06	844,09
Realización ensayos	65,00	12,06	783,80
Redacción memoria	55,00	12,06	663,22
Total	270,00 €		3.255,79 €

Tabla 10. Presupuesto mano de obra

A continuación se presenta el coste de los componentes empleados en el desarrollo del proyecto:

Descripción	Cantidad (ud)	Precio unitario (€/ud)	Subtotal (€)
Arduino UNO WIFI	1,00	51,40	51,40
Protoboard	1,00	1,66	1,66
Circuito integrado K3020P	1,00	26,00	26,00
Circuito integrado W08DC	1,00	7,06	7,06
Circuito integrado Pc 4N25	1,00	0,64	0,64
Circuito integrado TL431	1,00	0,74	0,74
Resistencia	5,00	2,18	10,90
Condensadores	2,00	0,50	1,00
Microswitch	1,00	1,20	1,20
Diodos	1,00	0,20	0,20
Cable USB AM/BM	1,00	3,98	3,98
Caja eléctrica FIBOX TEMPO	1,00	15,10	15,10
Cables	30,00	0,18	5,40
Total	47,00 €		125,28 €

Tabla 11. Presupuesto Coste de componentes utilizados

El coste de amortización del software utilizado en el proyecto:

Descripción	Coste unitario (€)	Tiempo amortización (años)	Coste amortización (€/h)	Cantidad (h)	Subtotal (€)
Licencia M. Office	399,00	3,00	0,07	270,00	19,95
Total	399,00 €				19,95 €

Tabla 12. Presupuesto Amortización Software

El coste de amortización de los equipos utilizado en el proyecto:

Descripción	Coste unitario (€)	Tiempo amortización (años)	Coste amortización (€/h)	Cantidad (h)	Subtotal (€)
Osciloscopio	503,19	5,00	0,06	175,00	9,78
Generador Ondas	197,23	5,00	0,02	175,00	3,84
Multímetro	125,00	5,00	0,01	175,00	2,43
PC Mc Book 13"	1499,00	4,00	0,21	270,00	56,21
Fuente Alimentación Variable	50,00	5,00	0,01	175,00	0,97
Total	2.374,42 €				16,05 €

Tabla 13. Presupuesto Amortización Equipos

El presupuesto de Ejecución Material:

Descripción	Coste (€)
Total Mano de Obra	3.255,79 €
Toral Componentes	125,28 €
Total Amortización Software	19,95 €
Total Amortización Equipos	16,05 €
Presupuesto Ejecución Material	3.417,07 €

Tabla 14. Presupuesto Ejecución Material

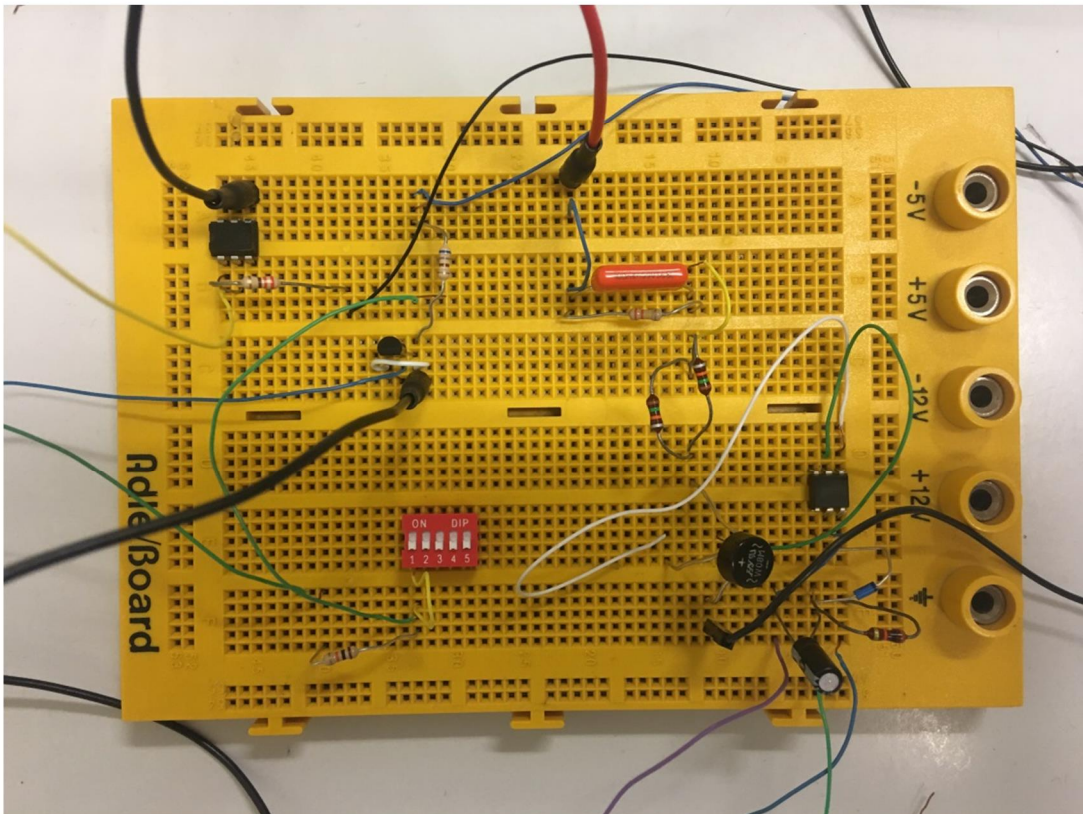
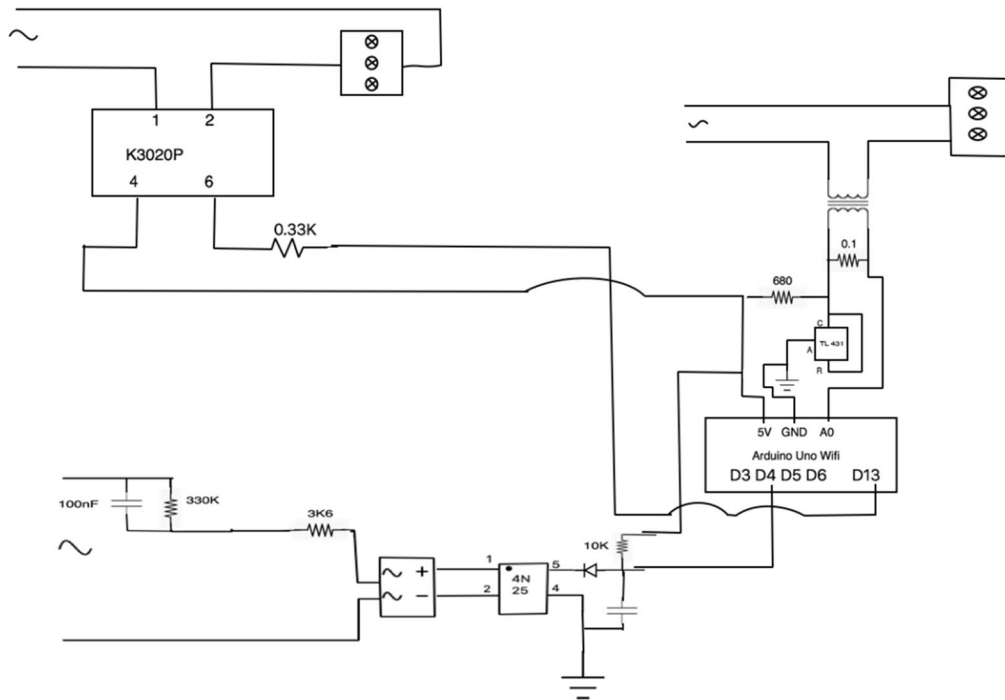
El presupuesto Total de Ejecución y el presupuesto Base de Licitación:

Descripción	Coste (€)
Presupuesto Ejecución Material	3.417,07 €
Gastos Generales (12%)	410,05 €
Beneficio Industrial (6%)	205,02 €
Presupuesto Total de Ejecución	4.032,14 €
IVA (21%)	846,7499344
Presupuesto Base de Licitación	4.878,89 €

Tabla 15. Presupuesto Total de Ejecución y Base de Licitación

El Presupuesto asciende a la cantidad de CUATRO MIL OCHOCIENTOS SETENTA Y OCHO EUROS CON OCHENTA Y NUEVE CÉNTIMOS.

ANEXO 1. Esquemas de montaje



ANEXO 2. Sketches de Programación

En el anexo de programación se van a detallar cada sketch que se ha utilizado tanto para la medida de los ensayos, como para el diseño final.

En primer lugar, se muestran los sketch realizados mediante el software Ide Arduino.

Sketch detección de estado:

```
boolean a; //Declaración variable booleana a
void setup() {
Serial.begin(115200); //Establecimiento velocidad de comunicación
pinMode(5,INPUT); //Configuración del pin 5 como entrada.
}

void loop() {
a=digitalRead(5); //lectura del pin digital 5
if(a==0){ //recibe un 0=0V
Serial.println ("el equipo esta encendido");
}
else { //Recibe un 1=5V
Serial.println("el equipo esta apagado");
}
delay(3000); //tiempo de espera para facilitar la lectura
}
```

Sketch lectura tensión continua:

```
//declaración de variables
int valor_sensor; //declaración de variables tipo entero
float voltaje; //Declaración de variables tipo

void setup() {
// establecer la velocidad de comunicación serie
Serial.begin(115200);
}

void loop() {

int valor_sensor = analogRead(A0); //lectura analógica pin analogico A0

voltaje = 0,959*(valor_sensor) * (5.0 / 1024.0); // Convierte la lectura analogica corregida
(0 - 1024) hacia un voltaje (0-5V):
```

```
Serial.println(voltaje);  
delay(3000);  
}
```

Sketch lectura de intensidad:

```
int valor_sensor;  
float voltaje;  
float voltajeficaz;  
float Vinst;  
float Vsum;  
int n;  
float intensidad;  
float Potapar;  
float a;  
int tension;  
void setup() {  
  Serial.begin(115200);  
}  
  
void loop() {  
  tension=230;  
  n=0;  
  Vinst=0;  
  Vsum=0;  
  while(n<81){  
    // lee el pin de entrada analogico (pin A0)  
  
    int valor_sensor = analogRead(A0);  
    // Convierto la lectura analogica (la cual va desde 0 - 1024) hacia un voltaje (0 - 5V):  
    voltaje = (valor_sensor-532) * (5.0 / 1024.0);  
    Vinst=voltaje*voltaje;  
    Vsum=Vsum+Vinst;  
    n++;  
    delay(0.4);  
  }  
  a=5.15;  
  voltajeficaz=sqrt(Vsum/n);  
  intensidad=a*voltajeficaz;  
  Potapar=intensidad*tension;  
  Serial.print("La intensidad es de ");  
  Serial.print(voltajeficaz);  
  Serial.println("A");
```

```
Serial.print("La tension es de ");
Serial.print(tension);
Serial.println("V");
Serial.print("La potencia aparente es de ");
Serial.print(Potapar);
Serial.println("VA");

    delay(3000);
}
```

Sketch diseño final:

```
int valor=0;
boolean a,b,c,d;
int luz=13;
boolean status=LOW; //Estado del led

void setup() {

    Serial.begin(115200);

    pinMode(3,INPUT);
    pinMode(4,INPUT);
    pinMode(5,INPUT);
    pinMode(6,INPUT);

    pinMode(luz,OUTPUT);

}

void loop() {
    //Lectura Digital
    a=digitalRead(3);
    b=digitalRead(4);
    c=digitalRead(5);
    d=digitalRead(6);
    if(a==true){ //si el valor leído es igual a 1 envia un 1 por el puerto serie
        valor=1;
        delay(150);
        Serial.write(valor);
    }
    else{
        valor=2;
        delay(150);
    }
}
```

```
Serial.write(valor);
}
if(b==true){
valor=4;
delay(150);
Serial.write(valor);
}
else{
valor=3;
delay(150);
Serial.write(valor);
}
if(c==true){
valor=5;
delay(150);
Serial.write(valor);
}
else{
valor=6;
delay(150);
Serial.write(valor);
}
if(d==true){
valor=7;
delay(150);
Serial.write(valor);
}
else{
valor=8;
delay(150);
Serial.write(valor);
}

if(Serial.available()>0)//Si el Arduino recibe datos a través del puerto serie
{
byte dato = Serial.read(); //Los almacena en la variable "dato"
if(dato==65) //Si recibe una "A" (en ASCII "65")
{
status=!status; //Cambia el estado del led
}
digitalWrite(led,status);
}
}
```

A continuación, se muestra el sketch realizado por el software Processing para la realización de la interfaz gráfica:

```

import processing.serial.*;

int valor;
PImage img,img1,img2,img3, img4, img5;
int intradquad=40; //Radio del interruptor
int intxquad=300; //Posición X de interruptor
int intyquad=200; //Posición Y de interruptor
boolean overRect = false; //Estado del mouse si está encima de rect o no
boolean status=false; //Estado del color de rect
String texto="Luz ON";//Texto del status inicial del LED
int red=0;//Colores del botón
int green=250;
int blue=0;

void setup() {
  size(1200, 800);// tamaño de pantalla
  String portName = Serial.list()[1]; //Muestra en consola lista de puertos disponibles
  myPort = new Serial(this, portName, 115200);// eleccion puerto de comunicacion
  textSize(12);
  img = loadImage("lavadora.jpg");//carga las imágenes
  img1 = loadImage("imac.jpg");
  img2 = loadImage("cadenademusica.jpg");
  img3 = loadImage("plancha.jpg");
  img4 = loadImage("upv.jpg");
  img5 = loadImage("etsii.jpg");
  fill(0,250,0);
  stroke(0,0,0);
  ellipse(300,450,10,10);
}

void draw() {

  fill(0,0,0);
  PFont f0 = loadFont("ArialNarrow-48.vlw");//Tipo de fuente
  textFont(f0, 40);
  text("Diseño de un equipo de supervisión de una instalación eléctrica", 140, 65);

  image(img, 190,310,80,80);
  image(img1, 330,310,80,80);
  image(img2, 190,510,80,80);
  image(img3, 330,510,80,80);

```



```
image(img4, 10,10,80,80);  
image(img5, 1110,10,80,80);
```

```
stroke(0,0,0); //Dibujo del plano  
fill(250,250,250);  
line(200,400,400,400);//cuadrado grande  
line(200,500,400,500);  
line(200,400,200,500);  
line(400,400,400,500);  
line(200,430,230,430);//Cuadrado pequeño ED1  
line(230,430,230,400);  
line(370,400,370,430);//Cuadrado pequeño ED2  
line(370,430,400,430);  
line(200,470,230,470);//Cuadrado pequeño ED3  
line(230,470,230,500);  
line(370,500,370,470);//Cuadrado pequeño ED4  
line(370,470,400,470);  
line(285,435,315,435);//Cuadrado pequeño centro  
line(315,435,315,465);  
line(315,465,285,465);  
line(285,465,285,435);  
if (myPort.available() > 0) { // si se recibe algun dato por el puerto de comunicacion lo  
lee  
    valor = myPort.read();  
  
    if (valor==1){//Aparato 1 encendido  
        Lav1=true;  
        fill(0,250,0);  
        stroke(0,0,0);  
        ellipse(215,415,10,10);  
  
    }  
  
    if(valor==2) {//Aparato 1 Apagado  
  
        fill(250,0,0);  
        stroke(0,0,0);  
        ellipse(215,415,10,10);  
  
    }  
}
```

```
}
```

```
if (valor==3){//Aparato 2 encendido  
  fill(0,250,0);  
  stroke(0,0,0);  
  ellipse(385,415,10,10);
```

```
}
```

```
if(valor==4) {//Aparato 2 Apagado  
  
  fill(250,0,0);  
  stroke(0,0,0);  
  ellipse(385,415,10,10);
```

```
}
```

```
if (valor==5){//Aparato 3 encendido  
  fill(0,250,0);  
  stroke(0,0,0);  
  ellipse(215,485,10,10);
```

```
}
```

```
if(valor==6) {//Aparato 3 Apagado  
  
  fill(250,0,0);  
  stroke(0,0,0);  
  ellipse(215,485,10,10);
```

```
}
```

```
if (valor==7){//Aparato 4 encendido  
  fill(0,250,0);  
  stroke(0,0,0);  
  ellipse(385,485,10,10);
```

```
}
```

```
if(valor==8) { //Aparato 4 Apagado

    fill(250,0,0);
    stroke(0,0,0);
    ellipse(385,485,10,10);

}

}
if(mouseX > intxquad-intradquad && mouseX < intxquad+intradquad && //Si el cursor
del raton se encuentra dentro de rect
mouseY > intyquad-intradquad && mouseY < intyquad+intradquad)
{
    overRect=true; //Variable que demuestra que el cursor del raton esta dentro de rect
    stroke(255,0,0); //Contorno de rect rojo
}
else
{
    overRect=false; //Si el cursor del raton no está dentro de rect, la variable pasa a ser
falsa
    stroke(0,0,0); //Contorno de rect negro
}

fill(red,green,blue); //Dibujamos un rectangulo de color gris
rectMode(RADIUS); //Esta función hace que Width y Height de rect sea el radio (distancia
desde el centro hasta un costado).
rect(intxquad,intyquad,intradquad,intradquad);
//Rect blanco

fill(0,0,0); //Creacion de un texto de color negro con la palabra Luz on
PFont f = loadFont("Calibri-48.vlw");//Tipo de fuente
textFont(f, 20);
text(texto, 270, 210);
PFont f1 = loadFont("Calibri-48.vlw");//Tipo de fuente
textFont(f1, 20);
text("luz", 290, 480);

}
```

```
void mousePressed() //Cuando el raton está apretado
{
  if (overRect==true) //Si el raton está dentro de rect
  {
    status=!status; //El estado del color cambia
    myPort.write("A"); //Envia una "A" para que el Arduino encienda el led
    if(status==true)
    {
      //Color del botón negro
      red=100;
      green=100;
      blue=100;
      texto="Luz OFF";
      fill(250,0,0);
      stroke(0,0,0);
      ellipse(300,450,10,10);
    }
    if(status==false)
    {
      //Color del botón verde
      red=0;
      green=250;
      blue=0;
      texto="Luz On";
      fill(0,250,0);
      stroke(0,0,0);
      ellipse(300,450,10,10);
    }
  }
}

void keyPressed()
{
  //Pulsar la tecla E para salir del programa
  if(key=='e' || key=='E')
  {
    exit();//Salimos del programa
  }
}
```

