



# Trabajo fin de grado de ingeniería de tecnologías de telecomunicación

## TÍTULO

Sistema fotovoltaico orientado a docencia

## AUTOR

Óscar Martínez Cano

## TUTOR

Miguel Ángel Carvajal Rodríguez

## COTUTOR

Antonio Martínez Olmos

## DEPARTAMENTO

Electrónica y Tecnología de Computadores







# Universidad de Granada

## AUTORIZACIÓN DE LECTURA

D. Miguel Ángel Carvajal Rodríguez, profesor del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada, como director del Trabajo Fin de Grado de D. Óscar Martínez Cano, informa que: El presente trabajo, titulado "Sistema fotovoltaico orientado a docencia" ha sido redactado por el mencionado alumno bajo mi dirección, y con esta fecha autorizo su presentación.

Granada, a 6 de Julio de 2015

Fdo. Miguel Ángel Carvajal Rodríguez





# Universidad de Granada

## AUTORIZACIÓN DE DEPÓSITO

Yo, Óscar Martínez Cano, autor del Trabajo Fin de Grado titulado "Sistema fotovoltaico orientado a docencia" realizado en la Universidad de Granada, autorizo al depósito de dicho Proyecto en la Biblioteca de la Universidad de Granada y/o Departamento de Electrónica y Tecnología de Computadores para ser libremente consultado por las personas a las que le resulte necesario.

Granada, a 6 de Julio de 2015

Fdo. Óscar Martínez Cano





# Universidad de Granada

## INFORME DE VALORACIÓN

El tribunal constituido para la evaluación del Trabajo Fin de Grado titulado:

"Sistema fotovoltaico orientado a docencia"

Realizado por el alumno:

Óscar Martínez Cano

Y dirigido por el tutor:

Miguel Ángel Carvajal Rodríguez

Ha resuelto asignarle la calificación de:

SOBRESALIENTE (9 - 10 puntos)

NOTABLE (7 - 8.9 puntos)

APROBADO (5 - 6.9 puntos)

SUSPENSO

Con la nota:                    puntos.

Tribunal:

Presidente: Carmen Benítez Ortúzar

Vocal 1: Juan José Ramos Muñoz

Vocal 2: Diego Pedro Morales Santos

Suplente: Victoria Sánchez Calle



## Agradecimientos

Quiero dedicarle este proyecto a mis padres por su apoyo incondicional y por ayudarme todos estos años de largo estudio para cumplir este sueño y que aún continúan haciéndolo, a mi hermano por animarme en los momentos difíciles y a todas las personas que lo hacen feliz a él, a mi pareja que ha estado ahí casi desde el principio, sufriendo conmigo en los momentos complicados y sin perder la sonrisa cuando hemos superado nuestras metas, a todos ellos por sentirlos cerca desde miles de kilómetros, a mis primos en los que siempre he confiado y he considerado mis mentores, que me han motivado para ser mejor persona y para tener un pensamiento crítico, a los amigos que me han dado calma en los momentos de estrés, y por último a los profesores, algunas de las personas más competentes que he conocido y que me han enseñado que aunque la vida te puede dar duras lecciones, siempre puedes recuperarte trabajando duro para conseguir tus objetivos.



# **Sumario**

## **Palabras clave**

Durante del desarrollo de este proyecto nos encontraremos con conceptos como: panel fotovoltaico, punto de máxima potencia, curva I-V, curva P-V, convertidor DC/DC elevador, inversor de puente completo , modulación PWM, Arduino, ciclo de trabajo, Hardware, Software, puerto serie o aplicación de PC.

## **Resumen**

Este proyecto está dedicado al desarrollo de un sistema fotovoltaico orientado a docencia, tanto de los componentes hardware como del software implementado y la interfaz de usuario, para obtener las curvas I-V y P-V de un panel fotovoltaico, el punto de máxima potencia y el factor de llenado. Para ello el sistema que vamos a realizar nos permitirá tomar medidas en tiempo real de la tensión y la corriente que proporciona el panel fotovoltaico, de manera local o remota, mediante un enlace de comunicaciones con un PC, al que se enviarán las muestras adquiridas con el convertidor A/D del microcontrolador. Para llevar a cabo todo esto aprenderemos técnicas de conmutación de señales de potencia, aprenderemos también a programar el microcontrolador ATMEGA 328P que lleva incorporado Arduino UNO, tanto utilizando algunas de las funciones que incorporan las librerías de Arduino, como a nivel de registro del microcontrolador, de esta manera comprenderemos mejor el funcionamiento interno de este dispositivo digital y obtendremos un código fuente optimizado. Por otro lado, también aprenderemos las características que tiene un enlace de comunicaciones serie como es el caso de una comunicación USB o Bluetooth y veremos algunas funciones de MATLAB que nos permiten comunicarnos con otros dispositivo, representar datos dinámicamente y crear un interfaz de usuario que simplifique el uso del dispositivo.

## **Aclaraciones**

Cada una de las imágenes de este documento que han sido obtenidas en Internet tiene una referencia numérica entre corchetes [N]. Por otro lado las referencias [CAR] indican que esa información ha sido toma de bibliografía de Miguel Ángel Carvajal Rodríguez. El listado de todas las referencias aparece al final de este documento.

Nota: Algunas referencias apuntan hacia foros de Internet no contrastados pero su contenido concuerda con los conocimientos adquiridos durante la carrera y la bibliografía consultada para la elaboración de este proyecto.



# **Índice de contenidos**

<b>1. Introducción</b>	19
1.1 Objetivo	20
1.2 Fundamento	20
1.3 Componentes del proyecto	20
1.4 Diagrama de bloques	21
<b>2. Energía solar fotovoltaica</b>	22
2.1 Motivaciones	23
2.2 El sol como fuente de energía	23
2.3 Conversión de la energía	23
2.4 Principios de funcionamiento de la energía fotovoltaica	24
2.4.1 Unión PN	24
2.4.2 Efecto fotoeléctrico	25
2.4.3 Funcionamiento de la célula fotovoltaica	26
2.4.4 Esquema eléctrico de un panel fotovoltaico	27
2.5 Corriente continua frente a corriente alterna	27
2.6 Tipos de instalaciones fotovoltaicas	28
2.7 Conexión a la red eléctrica	28
2.8 Ventajas e inconvenientes de la energía solar fotovoltaica	28
2.9 Mapa de radiación solar	29
<b>3. Caracterización de un panel fotovoltaico</b>	31
3.1 Introducción a las figuras de mérito de un panel fotovoltaico	32
3.2 Tipos de paneles fotovoltaicos	32
3.3 Factores de eficiencia de una célula solar	33
3.3.1 Punto de máxima potencia	33
3.3.2 Factor de llenado	34
<b>4. Desarrollo del convertidor elevador o Boost</b>	35
4.1 Introducción al convertidor elevador o <i>Boost</i>	36
4.2 Análisis del circuito	36
4.2.1 Modo de conducción continuo	37
4.2.2 Modo de conducción discontinuo	39
4.2.3 Límite de operación	40
4.3 Diseño del Boost	41
4.4 Simulación del convertidor elevador	41
4.5 Implementación física del convertidor <i>Boost</i>	42
4.6 Generación de la señal de control con Arduino	42
4.7 Acondicionamiento	44
4.7.1 Driver de potencia	44
4.7.2 Limitación de la tensión de salida del convertidor <i>Boost</i>	44
4.8 Sensores	45
4.8.1 Sensor de tensión	45
4.8.1.1 Divisor de tensión	45
4.8.1.2 Seguidor de tensión	46
4.8.1.3 Amplificador operacional LM358	46
4.8.2 Sensor de corriente	41
4.8.2.1 Amplificador operacional inversor	47
4.9 Simulación del convertidor <i>Boost</i> con los sensores	48
4.10 Calibración de los sensores	49
4.11 Medición del factor de llenado (Fill Factor)	50
4.12 Distribución de los componentes	51

<b>5. Desarrollo del inversor</b>	52
5.1 Introducción al inversor	53
5.2 Tipos de inversores	53
5.3 Inversor de puente completo	53
5.4 Análisis del circuito	54
5.5 Modulación por ancho de pulsos	55
5.5.1 Modulación bipolar	56
5.5.2 Modulación unipolar	57
5.6 Señales de control del puente inversor	59
5.6.1 Lógica digital necesaria para las señales de control	60
5.6.2 Circuito integrado CD74H00	61
5.7 Simulación del puente inversor	61
5.8 Filtro LC	65
5.8.1 Simulación del inversor con filtro LC	66
5.9 Generación de la señal PWM con Arduino	67
5.10 Precauciones con cargas inductivas	68
5.11 Señales de salida obtenidas con un carga resistiva	69
5.12 Señal a la salida del filtro LC	70
5.13 Distribución de los componentes	72
<b>6. Comunicación Arduino-PC. Aplicación de control</b>	73
6.1 Comunicación serie	74
6.2 Puerto Serie	74
6.2.1 USB (Universal Serial Bus)	75
6.3 USRAT (Universal Serial Asynchronous Receiver-Transmitter)	75
6.3.1 Comunicación USART	76
6.3.2 Interrupción por USART	76
6.4 Librería <i>serial</i> de Arduino	77
6.5 Comunicación serie con MATLAB	77
6.6 GUI de MATLAB	79
6.7 Diferentes caminos para crear de una GUI de MATLAB	79
6.8 Desarrollo de una interfaz de usuario	79
6.8.1 Entorno de desarrollo	79
6.8.2 Aplicación a desarrollar	81
6.9 Generación de un archivo ejecutable .EXE con deploytool	82
<b>7. Resultados</b>	84
<b>Apéndice I. Esquemático del circuito completo</b>	88
<b>Apéndice II. Líneas futuras. Transformador</b>	89
<b>Apéndice III. Líneas futuras. Almacenador</b>	90
<b>Apéndice IV. Acrónimos</b>	91
<b>Referencias</b>	92
<b>Bibliografía</b>	93



# Índice de figuras

<b>Figura 1.1:</b> Sistema fotovoltaico controlado con Arduino.	21
<b>Figura 2.1:</b> Esquema de funcionamiento de la unión PN.	24
<b>Figura 2.2:</b> Formación de la barrera interna de potencial.	24
<b>Figura 2.3:</b> Polarización directa e inversa respectivamente de la unión PN.	25
<b>Figura 2.4:</b> Emisión de electrones tras la absorción de la energía de un fotón.	25
<b>Figura 2.5:</b> Esquema eléctrico de un panel fotovoltaico.	27
<b>Figura 2.6:</b> Mapa de radiación solar en el mundo.	29
<b>Figura 2.7:</b> Mapa de radiación solar en España.	30
<b>Figura 2.8:</b> Invernaderos de Almería.	30
<b>Figura 3.1:</b> Distintos tipos de paneles solares.	32
<b>Figura 3.2:</b> Variación de la carga para obtener la curva I-V de un panel fotovoltaico.	33
<b>Figura 3.3:</b> Factores de eficiencia de una célula solar fotovoltaica.	33
<b>Figura 4.1:</b> Esquema de un convertidor <i>Boost</i> .	36
<b>Figura 4.2:</b> Distintos estados del convertidor <i>Boost</i> .	36
<b>Figura 4.3:</b> Formas de onda de corriente y voltaje del <i>Boost</i> en modo continuo.	37
<b>Figura 4.4:</b> Formas de onda de corriente y voltaje del <i>Boost</i> en modo discontinuo.	39
<b>Figura 4.5:</b> Convertidor <i>Boost</i> diseñado.	41
<b>Figura 4.6:</b> Tensiones y corrientes obtenidas de la simulación del convertidor <i>Boost</i> .	41
<b>Figura 4.7:</b> Señal de control y tensión en el drenador.	42
<b>Figura 4.8:</b> Diagrama temporal del temporizador 1 de Arduino en el modo CTC	43
<b>Figura 4.9:</b> Pines del driver de potencia del MOSFET.	44
<b>Figura 4.10:</b> Diodo zéner para limitar la tensión de salida del convertidor <i>Boost</i> .	45
<b>Figura 4.11:</b> Divisor de tensión.	46
<b>Figura 4.12:</b> Seguidor de tensión.	46
<b>Figura 4.13:</b> Configuración interna de los pines del circuito integrado LM358.	47
<b>Figura 4.14:</b> Amplificador operacional inversor.	47
<b>Figura 4.15:</b> Convertidor DC-DC elevador con los sensores de Vin e Iin.	48
<b>Figura 4.16:</b> Resultado de la simulación de los sensores de tensión y corriente.	49
<b>Figura 4.17:</b> Curva de calibración de la tensión.	49
<b>Figura 4.18:</b> Curva de calibración de la corriente.	50
<b>Figura 4.19:</b> Conexión del transistor que permite poner el circuito en abierto.	51
<b>Figura 4.20:</b> Situación de los componentes del convertidor DC-DC elevador.	51
<b>Figura 5.1:</b> Inversor de puente completo.	53
<b>Figura 5.2:</b> Posibles combinaciones de los interruptores de un puente completo.	54
<b>Figura 5.3:</b> Estados del inversor de puente completo.	54
<b>Figura 5.4:</b> Forma de onda cuadrada obtenida a la salida del puente.	55
<b>Figura 5.5:</b> Proceso de modulación PWM.	56
<b>Figura 5.6:</b> Modulación bipolar. Referencia y portadora. Salida.	56
<b>Figura 5.7:</b> Modulación unipolar. Referencias sinusoidales y portadora triangular. Salida.	57
<b>Figura 5.8:</b> Comparativa entre modulación bipolar y unipolar.	58
<b>Figura 5.9:</b> Cálculo de los ciclos de trabajo de trabajo.	58
<b>Figura 5.10:</b> Señales PWM, PWM filtrada y 50 Hz para el disparador del osciloscopio.	59
<b>Figura 5.11:</b> Zoom de la señal PWM.	59
<b>Figura 5.12:</b> PWM visualizada en el modo de alta resolución.	60
<b>Figura 5.13:</b> Lógica para la generación de las señales de conmutación del puente.	60
<b>Figura 5.14:</b> Diagrama funcional del CD74H00.	61

<b>Figura 5.15:</b> Esquemático del circuito del puente inversor.	62
<b>Figura 5.16:</b> Señales necesaria para comutar el puente.	63
<b>Figura 5.17:</b> Tensión a la salida del inversor.	63
<b>Figura 5.18:</b> Espectro de la tensión obtenida con una carga resistiva.	64
<b>Figura 5.19:</b> Función de transferencia del filtro LC.	65
<b>Figura 5.20:</b> Esquemático del inversor con filtro.	66
<b>Figura 5.21:</b> Señal obtenida a la salida del filtro.	66
<b>Figura 5.22:</b> Diagrama temporal del modo Fast PWM del timer 2 de Arduino.	67
<b>Figura 5.23:</b> Señales de control del puente.	68
<b>Figura 5.24:</b> Señales de control con tiempo de OFF.	69
<b>Figura 5.25:</b> Señal obtenida en la carga.	69
<b>Figura 5.26:</b> Zoom de la señal obtenida en la carga.	70
<b>Figura 5.27:</b> Señal obtenida a la salida del filtro.	71
<b>Figura 5.28:</b> Señal de salida del filtro LC mejorada.	71
<b>Figura 5.29:</b> Situación de los componentes del inversor.	72
<b>Figura 6.1:</b> Ejemplo de comunicación serie.	74
<b>Figura 6.2:</b> Patillaje USB A y USB B.	75
<b>Figura 6.3:</b> Enlace TTL.	75
<b>Figura 6.4:</b> Formato de la trama.	76
<b>Figura 6.5:</b> Diagrama de flujo de la aplicación de control	78
<b>Figura 6.6:</b> Ventana de desarrollo de GUIDE.	80
<b>Figura 6.7:</b> Ventana con los componentes necesarios para el proyecto.	80
<b>Figura 6.8:</b> Compilador de MATLAB.	82
<b>Figura 6.9:</b> Instalación de la aplicación.	83
<b>Figura 7.1:</b> Curvas I-V y P-I a las 12:00.	85
<b>Figura 7.2:</b> Curvas I-V y P-I a las 16:00.	85
<b>Figura 7.3:</b> Curvas I-V y P-I a las 18:30.	86
<b>Figura 7.4:</b> Interfaz de usuario operativa.	87



---

# **1. Introducción**

En este primer capítulo de introducción se va a hablar de los objetivos sobre los que se centra la realización de este proyecto y expondremos una visión global del funcionamiento del mismo. Para ello estudiaremos cada uno de los componentes que forman el dispositivo y los algoritmos desarrollados para su correcto funcionamiento y hablaremos sobre la utilidad de cada uno de ellos.

---

## **1.1 Objetivos**

En este proyecto se va a desarrollar un sistema de alimentación fotovoltaico orientado para su uso en docencia, y su principal objetivo es obtener el punto de máxima potencia de un panel fotovoltaico, que es una de las figuras de mérito que caracterizan a las células fotovoltaicas y que estudiaremos a fondo en el capítulo 3, y otro de los objetivos es obtener, a partir de la señal de continua que proporciona el panel solar, una señal de corriente alterna compatible con la mayoría de dispositivos que conectamos a la red eléctrica.

## **1.2 Fundamento**

El dispositivo estará formado por varios módulos sobre los que iremos hablando en los siguientes capítulos. En concreto, contaremos con un panel solar fotovoltaico, el cual genera una señal de corriente continua que será conectado a un convertidor DC-DC elevador, que es un dispositivo que nos permite variar la tensión y la corriente que proporciona el panel sin perder potencia en el proceso, de esta manera podremos calcular el punto de máxima potencia de dicho panel. Una vez tengamos el sistema operando a su máxima potencia, la señal obtenida pasará por un inversor, que es un convertidor DC-AC que nos proporcionará una señal de corriente alterna. Para controlar todo el proceso tendremos que implementar una serie de algoritmos que generarán las señales de control de las diferentes etapas y que nos permitirán tomar medidas del estado del sistema en tiempo real. Para llevar a cabo tales acciones haremos uso de la plataforma de prototipado Arduino, que además nos servirá para establecer comunicación con un PC.

## **1.3 Componentes del proyecto**

El proyecto en su totalidad está compuesto por los siguientes módulos y algoritmos:

- Placa solar fotovoltaica MSX-01: Es el elemento que proporcionará la energía a la parte de potencia de nuestro dispositivo. Los componentes digitales e integrados serán alimentados externamente mediante la conexión USB del PC.
- Convertidor DC-DC elevador: Este el dispositivo que nos permitirá ajustar el panel fotovoltaico para que opere a máxima potencia, de forma relativamente eficiente, como se mostrará más adelante.
- Algoritmo para calcular el punto de máxima potencia: Para ajustar el convertidor tendremos que desarrollar un algoritmo de control que se base en las medidas de la tensión y la corriente de entrada al convertidor.
- Inversor de puente completo: Este etapa convertirá la señal de continua de salida del convertidor en una señal de alterna con la misma tensión de pico.
- Generador de señales PWM para conmutar el puente inversor: Para conseguir que una señal senoidal a la salida del inversor usaremos señales de control moduladas por ancho de pulso cuya información será la señal senoidal.

- Filtro LC para obtener una señal sinusoidal: Tras realizar la conmutación del puente obtendremos una señal de PWM oscilante que tendremos que filtrar para conseguir la señal de 50 Hz senoidal. Pero este filtrado no debe perder potencia por ello utilizaremos un circuito resonante LC.
- Arduino: Para la parte de control utilizaremos la plataforma de prototipado Arduino UNO, que tiene un microcontrolador ATMEGA 328P.
- Comunicación PC-Arduino: Con el fin de poder visualizar los datos que mide Arduino, estableceremos una comunicación serie entre PC y Arduino de manera que podremos ajustar el sistema de manera remota.
- Interfaz de usuario: Para facilitar el manejo del dispositivo crearemos una pequeña interfaz de usuario con MATLAB que podremos instalar en cualquier PC.

## 1.4 Diagrama de bloques

Arduino es una plataforma electrónica open-source. Con este dispositivo controlaremos la frecuencia de conmutación del convertidor elevador para obtener la máxima potencia que es capaz de proporcionar el panel fotovoltaico, lo que conseguiremos haciendo medidas de la corriente y la tensión de entrada con unos sensores que desarrollaremos y sobre los que hablaremos más adelante. Y por otra parte también generaremos las señales de control que se encargarán de la conmutación del puente inversor. En la siguiente figura podemos ver el diagrama de bloques del sistema.

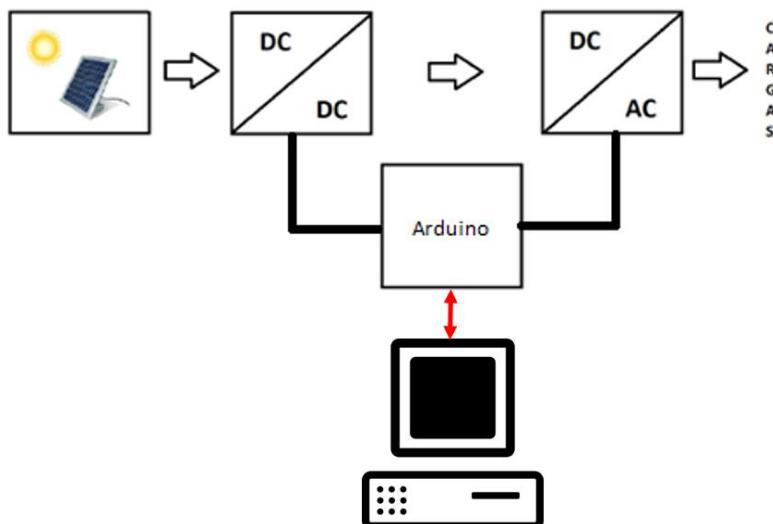


Figura 1.1: Sistema fotovoltaico controlado con Arduino.

En los próximos capítulos de este texto iremos explicando cómo se han desarrollado cada uno de los componentes del sistema, su funcionamiento, características y utilidades, desde los componentes más pequeños, pasando por cada una de las etapas del proyecto, hasta el software de control y la comunicación con el PC para garantizar que el lector comprenda lo que aquí estamos tratando.

---

## **2. Energía solar fotovoltaica**

En este capítulo se expone una visión general la energía solar fotovoltaica, se definen los conceptos de energía y potencia y se aplican al ámbito de la energía solar, se explican también la tecnología que se utiliza para convertir la energía solar en energía eléctrica y para transportarla, y los principios físicos que fundamentan todos estos procesos.

---

## 2.1 Motivaciones

La energía se define como la capacidad para realizar un trabajo y se denota en julios (J), mientras que la potencia es una medida de transferencia de energía a través del tiempo y se mide en vatios (W). La escala de Kardashov [1] es un método para medir el grado de evolución tecnológica de una civilización, basadas en la cantidad de energía utilizable y que se divide en tres categorías:

- **Tipo I** – Una civilización que es capaz de aprovechar toda la potencia disponible en un único planeta, aproximadamente  $10^{16}W$ .
- **Tipo II** – Una civilización que puede aprovechar toda la potencia disponible de una única estrella, aproximadamente  $10^{26}W$ .
- **Tipo III** – Una civilización que es capaza de aprovechar toda la potencia disponible de una sola galaxia, aproximadamente  $10^{37}W$ .

Kardashov definió originalmente el tipo I como el “nivel tecnológico cercano al nivel presente hoy en día en la Tierra”. Si Kardashov está en lo cierto, y teniendo unos recursos limitados en la Tierra, que además perjudican el medio ambiente, entonces ya podría ser el momento de pegar el salto hacia la civilización de tipo II.

## 2.2 El sol como fuente de energía

El Sol irradia en un segundo más energía que la consumida por la humanidad en toda su historia. Bajo la perspectiva humana, el Sol se puede considerar una fuente de energía inagotable. Esta energía proviene del proceso de fusión que se produce en el núcleo del Sol, donde los átomos de Hidrógeno se fusionan debido a la enorme fuerza de la gravedad, que los comprime hasta producir Helio. Se calcula que cada segundo, cerca de  $6 \cdot 10^{11}Kg$  de Hidrógeno son convertidos en Helio con una pérdida de masa neta de  $4 \cdot 10^3Kg$  que se convierten en luz y calor, siguiendo la ecuación de Einstein  $E = m \cdot c^2$ , en  $4 \cdot 10^{20}J$ . De esta manera el Sol permanece en equilibrio entre la fuerza de la gravedad que lo contrae y la energía generada en su interior que lo expande [2].

## 2.3 Conversión de la energía

La conversión fotovoltaica se basa en el efecto fotoeléctrico, el cual consiste en la emisión de electrones de un material cuando incide sobre él una radiación electromagnética como puede ser la luz visible o la luz ultravioleta. Para llevar a cabo esta conversión se utilizan unos dispositivos denominados células solares, constituidos por materiales semiconductores. Estas células conectadas en serie o paralelo forman un panel solar encargado de suministrar la tensión y la corriente que se ajuste a la demandada por el dispositivo conectado.

## 2.4 Principios de funcionamiento de la energía fotovoltaica

### 2.4.1 Unión PN

Las células solares están por lo general fabricadas en silicio, formando una gran superficie de unión PN [3]. En la siguiente figura podemos ver un ejemplo de su funcionamiento. Dicha unión está formada por la adhesión metalúrgica de dos cristales, generalmente de silicio (Si), aunque también se fabrican de Germanio (Ge), que son dopados con impurezas para modificar su conductividad, de manera que la zona N queda con exceso de electrones y la zona P queda con exceso de huecos.

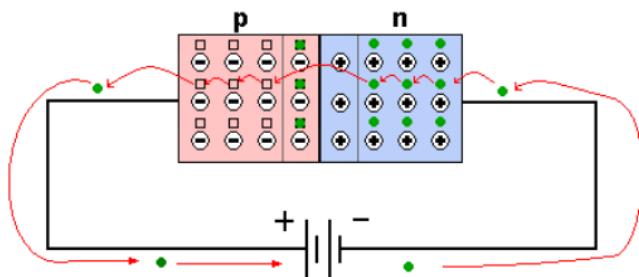


Figura 2.1: Esquema de funcionamiento de la unión PN. [4]

Al unir ambos cristales, se manifiesta una difusión de electrones del cristal N al cristal P, y al establecerse estas corrientes aparecen cargas fijas en una zona a ambos lados de la unión, esta zona recibe el nombre de barrera interna de potencial, cuya formación podemos ver en la figura siguiente.

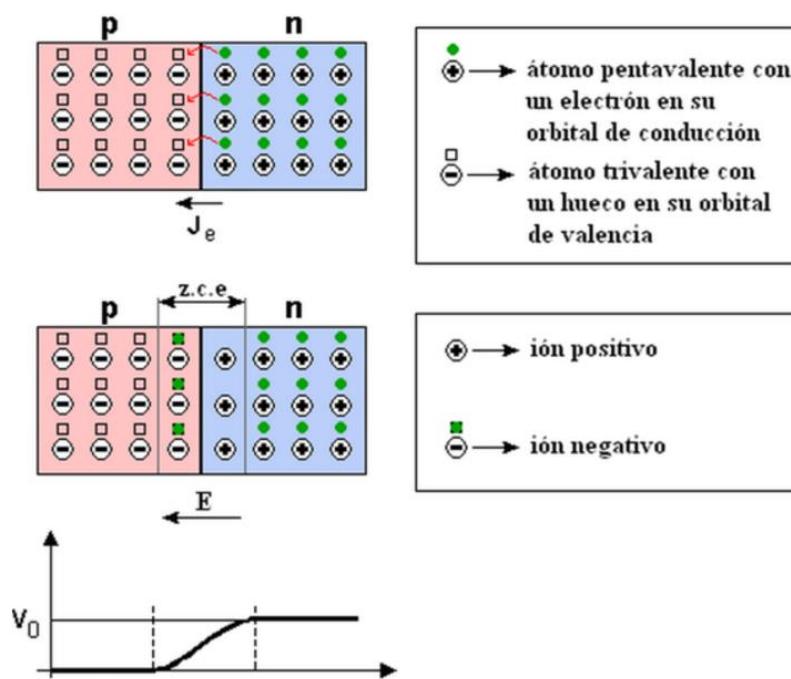


Figura 2.2: Formación de la barrera interna de potencial. [4]

Esta separación de cargas de una zona a la otra, creada por difusión, genera un campo eléctrico que actuará sobre los electrones libres de la zona N, con una fuerza de desplazamiento que se opondrá a la corriente de electrones y terminará deteniéndolos. Este campo eléctrico es equivalente a decir que aparece una diferencia de tensión entre las zonas P y N que denominamos ( $V_\gamma$ ), y que en el caso del Silicio es de  $0,7\text{ V}$  o  $0,3\text{ V}$  si los cristales son de germanio.

Cuando la unión se polariza directamente, la diferencia de potencial disminuye la barrera de potencial, permitiendo el paso de la corriente de electrones a través de la unión, mientras que si la unión se polariza inversamente, aumenta la barrera de potencial hasta alcanzar el valor de la diferencia de tensión aplicada y el diodo no debería conducir la corriente. A continuación podemos ver una imagen de ambos modos de funcionamiento:

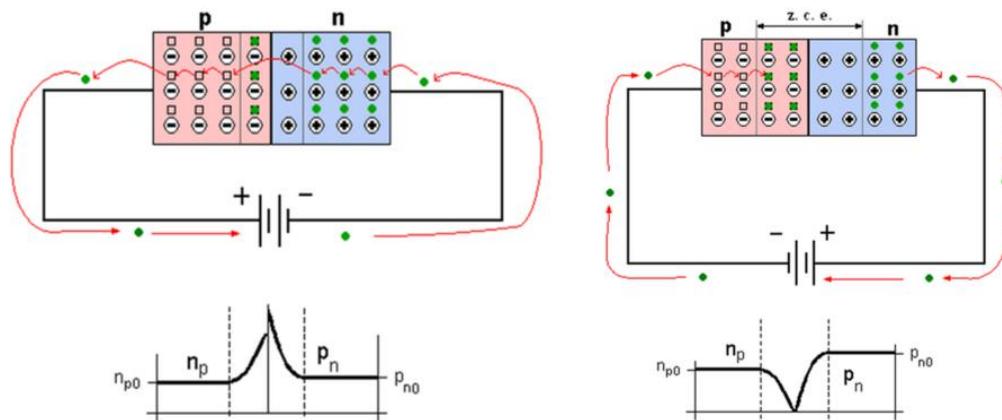


Figura 2.3: Polarización directa e inversa respectivamente de la unión PN. [4]

En la figura anterior podemos ver que cuando la unión se polariza directamente, hay un flujo de electrones a través de la unión, mientras que si la polarizamos inversamente aumenta la barrera de potencial y no hay flujo de corriente a través de la unión.

#### 2.4.2 Efecto fotoeléctrico

El efecto fotoeléctrico [5] consiste en la emisión de electrones por parte de un material cuando se hace incidir sobre él una radiación electromagnética. Véase la figura siguiente:

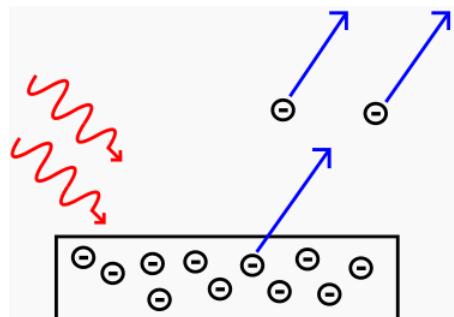


Figura 2.4: Emisión de electrones tras la absorción de la energía de un fotón. [6]

Los fotones tienen una energía característica determinada por la frecuencia de la luz. Si un átomo absorbe energía de un fotón y tiene más energía que la necesaria para expulsar un electrón del material y además posee una trayectoria dirigida hacia la superficie, entonces el electrón puede ser expulsado del material, por el contrario, si la energía del fotón es demasiado pequeña, el electrón es incapaz de escapar de la superficie del material. Esto es, la energía de un fotón absorbido necesaria para que un electrón pueda escapar del material es igual a la energía necesaria para liberar dicho electrón más su energía cinética. Algebraicamente se puede expresar como:

$$hf = hf_0 + \frac{1}{2}mv_m^2$$

O de una manera más compacta:

$$hf = \phi + E_k$$

Donde:

$h$  = Constante de Planck =  $6,63 \cdot 10^{-34}$  Js.

$f_0$  = Frecuencia mínima del fotón para que tenga lugar el efecto fotoeléctrico (Hz).

$\phi$  = Función trabajo o mínima energía necesaria para llevar un electrón del nivel de Fermi al exterior del material.

$E_k$  = Máxima energía cinética de los electrones que se observa experimentalmente.

Entonces:

- Si  $h \cdot f < \phi$  los electrones no obtienen suficiente energía como para ser emitidos de la superficie metálica.
- Si  $h \cdot f \geq \phi$  la energía del fotón puede ser absorbida produciendo un salto desde la banda de valencia a la banda de conducción, dejando un hueco en la banda de valencia y generando por tanto un par electrón -hueco.

#### 2.4.3 Funcionamiento de una célula fotovoltaica

Si ya hemos conseguido entender los conceptos previamente introducidos sobre la física de semiconductores, unión PN y efecto fotoeléctrico, el principio de funcionamiento de un panel fotovoltaico podemos describirlo de una manera sencilla de la siguiente manera:

Algunos de los fotones, que provienen de la radiación solar, impactan sobre la superficie del panel, penetrando en este y siendo absorbidos por materiales semiconductores, tales como el silicio o el arseniuro de galio. Por otro lado los electrones, que son partículas subatómicas que forman parte del exterior de los átomos, y que se alojan en orbitales de energía cuantificada, son golpeados por los fotones liberándose de los átomos que los mantenían confinados. Esto les permite, posteriormente, circular a través del material produciendo electricidad. Por otro lado, las cargas positivas complementarias que se crean en los átomos que pierden los electrones (huecos), fluyen en sentido contrario al de los electrones, en el panel solar. En palabras más sencillas, el panel solar convierte la energía solar (energía en forma de radiación que depende de la frecuencia de los fotones) en una determinada cantidad de corriente continua.

#### 2.4.4 Esquema eléctrico de un panel fotovoltaico

Ya que conocemos el principio de funcionamiento de la célula fotovoltaica, lo siguiente que vamos a ver es su modelo eléctrico. Para ello fijémonos en la siguiente figura:

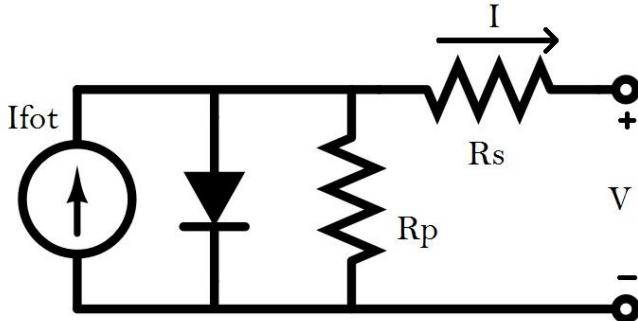


Figura 2.5: Esquema eléctrico de un panel fotovoltaico. [7]

En esta figura podemos ver un modelo matemático equivalente a una única célula solar individual, donde  $I$  y  $V$  son respectivamente la corriente y el voltaje generados,  $R_p$  representa la caída de potencial debida a los contactos del panel,  $R_s$  representa las corrientes de fuga a través de la unión, la unión se representa con un diodo e  $I_{fot}$  es la corriente producida por el efecto fotovoltaico. La tensión y la corriente de salida son señales continuas, y este tipo de señal pierde potencia durante la transmisión y además la mayoría de dispositivos electrónicos de consumo funcionan con corriente alterna, como lámparas, o realizan una conversión AC-DC, como los cargadores de los teléfonos móviles. Veamos algunas de las diferencias entre la corriente continua y la corriente alterna.

#### 2.5 Corriente continua frente a corriente alterna

El hecho de que la red eléctrica funcione con corriente alterna se debe a que la transmisión de corriente continua es poco adecuada para grandes distancias y amplitudes, ya que sufre enormes pérdidas por disipación de calor según el principio de Joule [8]. Este fenómeno consiste en que si por un conductor circula corriente eléctrica, parte de la energía cinética de los electrones se transforma en calor debido a los choques que sufren con los átomos del material conductor por el que circulan, elevando la temperatura del mismo. Este efecto es utilizado para calcular la energía disipada en un conductor atravesado por una corriente eléctrica, de la siguiente manera:

$$\begin{cases} P = V \cdot I \\ E = P \cdot t \end{cases} \rightarrow E = V \cdot I \cdot t$$

La potencia  $P$  disipada en un conductor es igual a la diferencia de potencial  $V$  a la que está sometido multiplicada por la intensidad de corriente  $I$  que lo atraviesa. La energía  $E$  es el producto de la potencia “ $P$ ” por el tiempo “ $t$ ” transcurrido. Luego la energía “ $E$ ” es el producto de la tensión “ $V$ ” por la intensidad “ $I$ ” y por el tiempo “ $t$ ”. Si añadimos la Ley de Ohm, la expresión queda de la siguiente manera:

$$\begin{cases} E = V \cdot I \cdot t \\ I = \frac{V}{R} \end{cases} \rightarrow E = \frac{V^2}{R} \cdot t$$

Si analizamos el problema de cerca vemos que las pérdidas en la transmisión de energía dependen directamente de la corriente, pero también vemos que la corriente y la tensión se relacionan inversamente con la potencia, es decir, a mayor voltaje, menor intensidad de corriente es necesaria para transmitir la misma potencia y por lo tanto, menores pérdidas. Y a diferencia de las señales DC, el voltaje de las señales AC se puede elevar, disminuyendo la corriente en igual proporción, con un transformador para ser transportado largas distancia con pocas pérdidas, y antes de proveer energía a los clientes, el voltaje se puede reducir a niveles seguros y económicos.

## 2.6 Tipos de instalaciones fotovoltaicas

En este apartado hablaremos sobre los diferentes tipos de instalaciones fotovoltaicas:

- Instalaciones aisladas: La energía se utiliza para cubrir pequeños consumos eléctricos en el mismo lugar donde se produce la demanda, como: viviendas alejadas, alumbrado público, aplicaciones agrícolas, señalización, etc.
- Instalaciones conectadas a la red eléctrica. En este caso parte de la energía se consume y el resto se entrega a la red eléctrica.

Nuestro proyecto sería algo parecido a una instalación aislada. Pero para ello deberíamos disponer de un sistema de almacenaje de energía y un transformador. Pero esto lo dejaremos para líneas futuras.

## 2.7 Conexión a la red eléctrica

Un aspecto importante en los sistemas fotovoltaicos aislados comerciales es que la frecuencia inyectada debe ser igual a la frecuencia de la red, que en Europa son 50 Hz. Pero esta frecuencia sufre pequeñas variaciones debido a perturbaciones en la red, por esta razón se ha de realizar una lectura en tiempo real de la frecuencia de la red para así poder regular la conmutación del puente inversor y tener la misma frecuencia que la frecuencia instantánea de la red eléctrica. En nuestro caso, al tratarse de un proyecto orientado a docencia, no tendremos en cuenta las variaciones de la red, sino que generamos una señal de 50 Hz de manera local con Arduino.

## 2.8 Ventajas e inconvenientes de la energía solar fotovoltaica

A continuación vamos a ver cuáles son algunas ventajas e inconvenientes del uso de energía solar fotovoltaica.

### Ventajas

- No produce combustión, por lo que no se generan contaminantes atmosféricos
- El silicio es muy abundante
- No requiere grandes tendidos eléctricos.

## Inconvenientes

- Impacto en el proceso de fabricación: Extracción del Silicio, fabricación de las células
- Explotaciones conectadas a red: Necesidad de grandes extensiones de terreno. Impacto visual.

## 2.9 Mapa de radiación solar

Para terminar este capítulo vamos a ver cómo de rentable es el uso de la energía solar fotovoltaica [9]. En este sentido el mapa de radiación solar es muy importante porque nos muestra las partes del mundo dónde se puede aprovechar mejor la energía solar. Un dato curioso es que los países donde hay menor radiación de energía solar es donde más se está usando este tipo de energía. Alemania por ejemplo es pionera en el uso de energía solar fotovoltaica y recibe menos de 150 W/m<sup>2</sup>. En el sur de España podemos llegar a recibir 300W/m<sup>2</sup>.

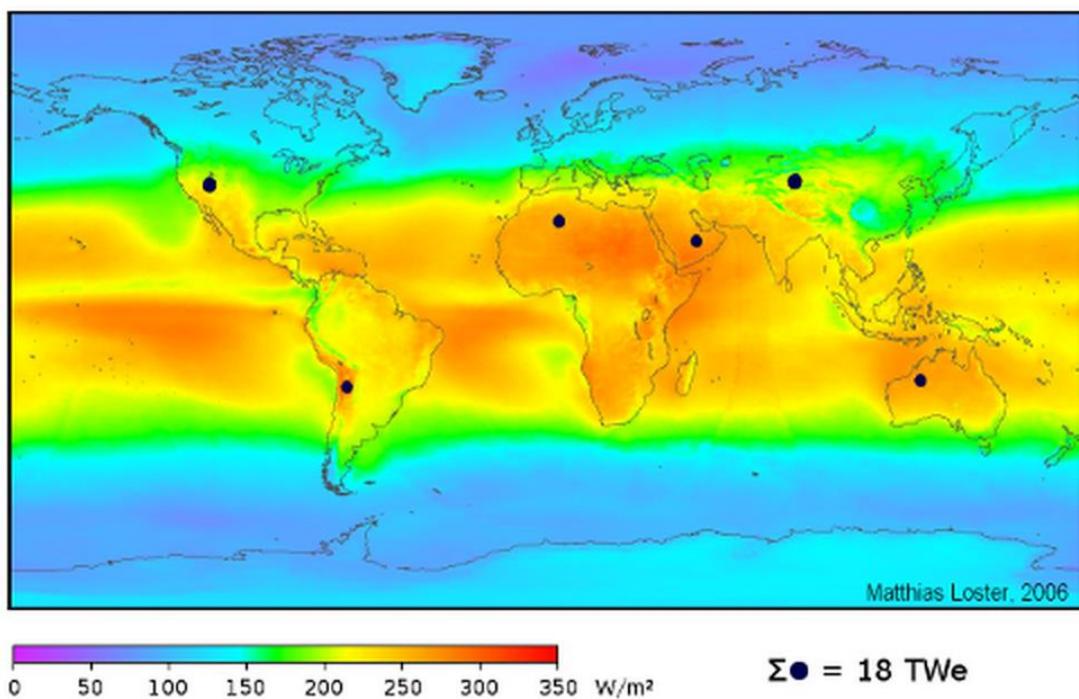


Figura 2.6: Mapa de radiación solar en el mundo. [10]

En la siguiente figura podemos ver más concretamente en qué situación se encuentra España en cuanto a los niveles de radiación solar que recibe se refiere.

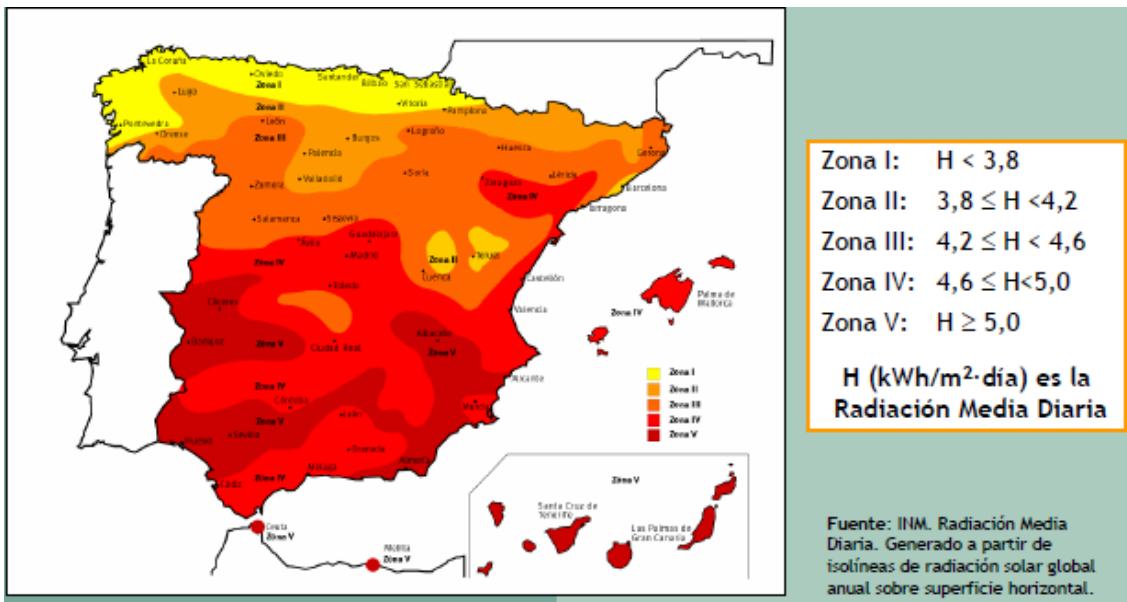


Figura 2.7: Mapa de radiación solar en España. [11]

Como curiosidad, comentar que si todos los invernaderos de Almería fueran paneles solares, producirían la energía necesaria para abastecer toda Andalucía. Y además los invernaderos no tendrían que dejar de ser invernaderos si la tecnología experimental de paneles solares transparentes de perovskita sale adelante.



Figura 2.8: Invernaderos de Almería. [15]

---

### **3. Caracterización de un panel fotovoltaico**

En este capítulo vamos a aprender cuales son las figuras de mérito de un panel fotovoltaico y vamos a describir cómo obtenerlas e interpretarlas para conseguir el mayor rendimiento posible de un panel fotovoltaico

---

### 3.1 Introducción a las figuras de mérito de un panel fotovoltaico

Los paneles fotovoltaicos [16] están formados por un conjunto de células fotovoltaicas que producen electricidad a partir de la luz que incide sobre ellos, y el parámetro estandarizado para clasificar se denomina potencia de pico, que se corresponde con la potencia máxima que el módulo puede entregar bajo unas condiciones estandarizadas que son:

- Radicación de  $1000 \frac{W}{m^2}$
- Temperatura de célula  $25^\circ C$

Además hay otros parámetros que nos permiten caracterizar la eficiencia de un panel fotovoltaico, como la eficiencia de la conversión de energía, el factor de llenado y la temperatura de operación nominal de la célula. Estos valores pueden variar de un panel fotovoltaico a otro, bien por sus dimensiones o bien por cómo están construidos.

### 3.2 Tipos de paneles fotovoltaicos

Los paneles fotovoltaicos, en función del tipo de célula que los forman, se dividen en:

- Cristalinos: Se dividen en dos categorías.
  - Monocristalinos: Se componen de secciones de un único cristal de silicio (Si).
  - Policristalinos: Están formadas por pequeñas partículas cristalizadas.
- Amorfos: Son aquellas en las que el silicio se dispone formando una red cristalina.

En cuanto a las características de cada uno de estos paneles, decir que el rendimiento de los paneles fotovoltaicos cristalinos suele ser del 20% mientras que los amorfos solo llegan al 10%, sin embargo, el coste y peso de estos últimos es muy inferior. El panel MSX-01 que utilizamos en nuestro proyecto es un panel policristalino. En la siguiente figura podemos ver el aspecto que tiene cada uno de estos paneles:

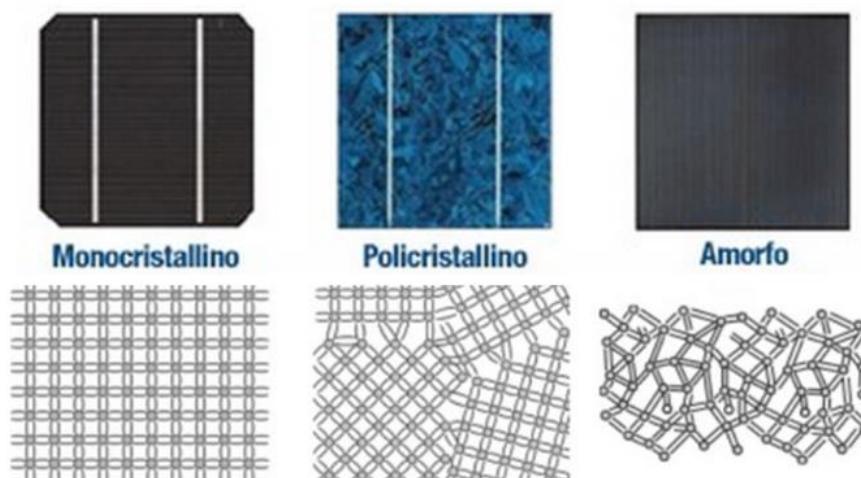


Figura 3.1: Distintos tipos de paneles solares. [17]

### 3.3 Factores de eficiencia de una célula solar

En este apartado vamos a ver cuáles son las figuras de mérito que vamos obtener para caracterizar nuestro panel fotovoltaico.

#### 3.3.1 Punto de máxima potencia

Una placa o célula solar puede operar en un amplio rango de voltajes e intensidades de corriente. Esto puede lograrse variando la impedancia de la célula desde el valor 0 (cortocircuito) a valores muy altos (circuito abierto) y se puede determinar el punto de potencia máxima teórica, es decir, el estado del circuito en el que se maximiza la tensión y la corriente, o lo que es lo mismo, la carga para la cual la célula puede entregar la máxima potencia eléctrica. Para comprender mejor este concepto podemos fijarnos en la siguiente figura.

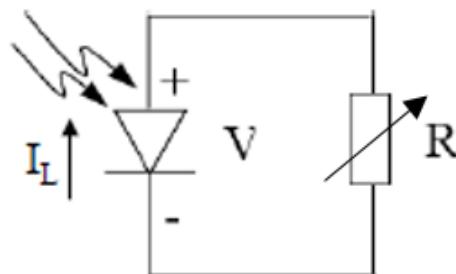


Figura 3.2: Variación de la carga para obtener la curva I-V del panel fotovoltaico. [CAR]

El punto de potencia máxima de un panel fotovoltaico varía con la iluminación incidente, por lo que un ajuste dinámico de la carga del sistema hará que el panel opere a máxima potencia, lo que producirá un gran incremento en la eficiencia de nuestro dispositivo. Si en el sistema que nos muestra la figura 3.2 variamos el valor de la carga obtendremos diferentes valores de tensión y corriente. En la figura siguiente podemos ver el resultado de la variación de la carga, la curva I-V de un panel fotovoltaico y la curva P-V.

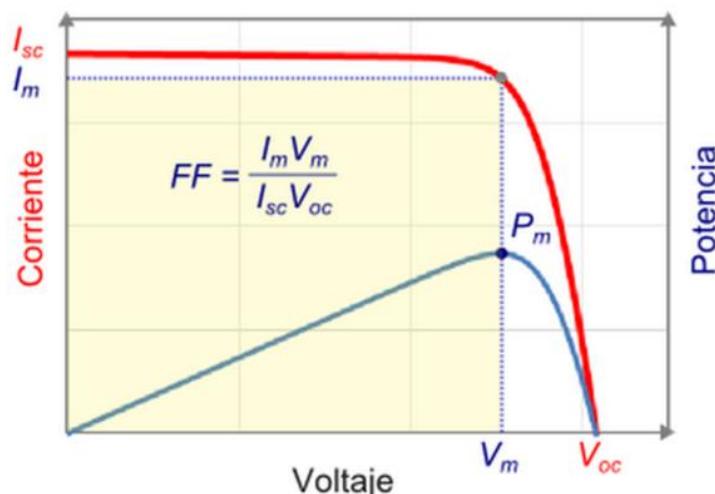


Figura 3.3. Factores de eficiencia de una célula solar fotovoltaica. [18]

Si nos fijamos en la figura anterior, por un lado tenemos  $I_{sc}$  y  $V_{oc}$ , que son los valores de la corriente en cortocircuito y la tensión en circuito abierto respectivamente y, por otro lado,  $I_m$  y  $V_m$ , que son los valores de corriente y tensión para los que la potencia suministrada por el panel es máxima. La relación entre estos valores nos da el factor de llenado.

### 3.3.2 Factor de llenado

El factor de llenado o fill factor (FF) es otro término para definir la eficacia de una célula solar y se define como la relación entre la potencia máxima entre el voltaje en circuito abierto ( $V_{oc}$ ) y la corriente en cortocircuito ( $I_{sc}$ ). El producto de estos dos valores representa la potencia máxima ideal, si el panel no tuviera pérdidas. Matemáticamente se puede expresar como:

$$FF = \frac{P_m}{V_{oc} \cdot I_{sc}}$$

Unos de nuestros objetivos en este proyecto será conseguir que el panel opere a máxima potencia y calcular el factor de llenado, y estos datos serán enviados y representados en un PC.

En nuestro caso la carga que conectaremos al panel fotovoltaico será el convertidor DC-DC elevador, que es un dispositivo que nos permite variar la tensión y la corriente sin perder potencia en el proceso. Dicha variación se realizará mediante el control del ciclo de trabajo de una señal PWM que generaremos con Arduino. En el siguiente capítulo veremos cómo funciona un convertidor DC-DC elevador y cuáles son sus características que nos permitirán cumplir este objetivo.

---

## **4. Desarrollo del convertidor DC-DC elevador**

En este capítulo se exponen las técnicas utilizadas para la implementación del convertidor DC-DC elevador, así como los resultados obtenidos tanto en la simulación de esta etapa con PSPICE, como los resultados experimentales medidos en el laboratorio, además se dará una explicación detallada de por qué se utiliza este módulo en nuestro proyecto de manera que entendamos cuál es la relación que tiene este dispositivo con la obtención del punto de máxima potencia.

---

## 4.1 Introducción al convertidor DC-DC elevador o *Boost*

Un convertidor *Boost* [19] es un convertidor DC a DC que obtiene a su salida un nivel de tensión continua mayor que en su entrada. Es un tipo de fuente conmutada que contiene al menos dos dispositivos semiconductores (diodo y transistor), y al menos un elemento para almacenar energía (condensador, bobina o combinación de ambos). Su utilidad radica en que algunas fuentes DC como baterías o paneles solares pueden producir niveles de tensión inadecuados y este dispositivo se puede ajustar para conseguir un nivel de voltaje más adecuado. En la figura siguiente podemos ver un esquema básico de un convertidor elevador

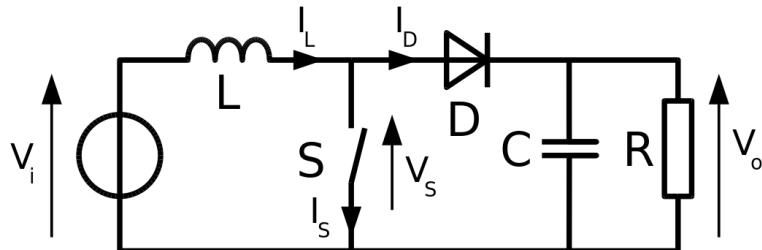


Figura 4.1: Esquema de un convertidor *Boost*. [20]

El elevador que vamos a implementar conmuta la energía desde una bobina a un condensador gracias a un transistor que utilizaremos con interruptor y un diodo que impide que la energía almacenada en el condensador vuelva a la bobina. A primera vista puede parecer un circuito sencillo pero tiene algunas peculiaridades y funciones que nos permitirán obtener la máxima potencia del panel solar y que explicaremos en el siguiente apartado.

## 4.2 Análisis del circuito

El principio básico del convertidor *Boost* consiste en dos estados distintos dependiendo de la posición del interruptor; cuando el interruptor está cerrado, la bobina almacena energía de la fuente y la carga queda alimentada por el condensador de salida y, cuando el interruptor está abierto, la corriente fluye a través del diodo hacia el resistor y el condensador de salida, el cual quedará cargado completamente.

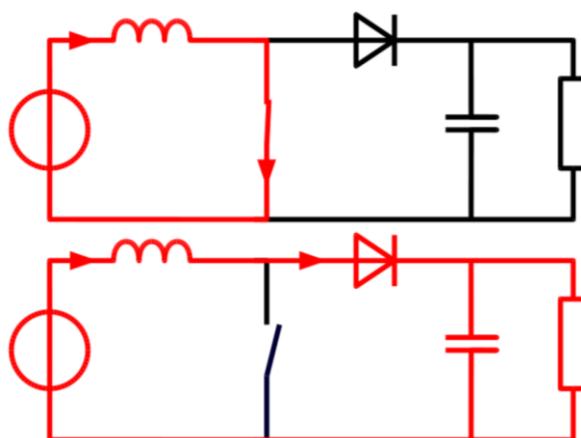


Figura 4.2: Distintos estados del convertidor *Boost*. [20]

Este proceso de conmutación de energía entre la bobina y el condensador es controlado mediante la frecuencia y ciclo de trabajo de la señal que controla el estado del interruptor, de esta manera con una señal de control adecuado obtendremos el nivel de tensión y de corriente que maximiza la potencia que puede entregar el panel, en otras palabras, cuando alimentemos este dispositivo con el panel solar y ajustemos la señal de control podremos obtener las figuras de mérito que comentamos en el capítulo anterior.

Para conseguir que el elevador funcione correctamente tenemos que saber los distintos modos de funcionamiento de este dispositivo: Modo continuo, en el que toda la energía se transfiere a la carga sin que la corriente a través del inductor llegue a anularse, y Modo discontinuo, en el cual la carga consume menos potencia de la que el circuito puede suministrar en un periodo de conmutación.

#### 4.2.1 Modo de conducción continuo

En este modo de operación, la corriente a través de la bobina nunca llega a cero. La siguiente figura muestra las formas de ondas típicas de la corriente y el voltaje en el convertidor cuando está operando en este modo.

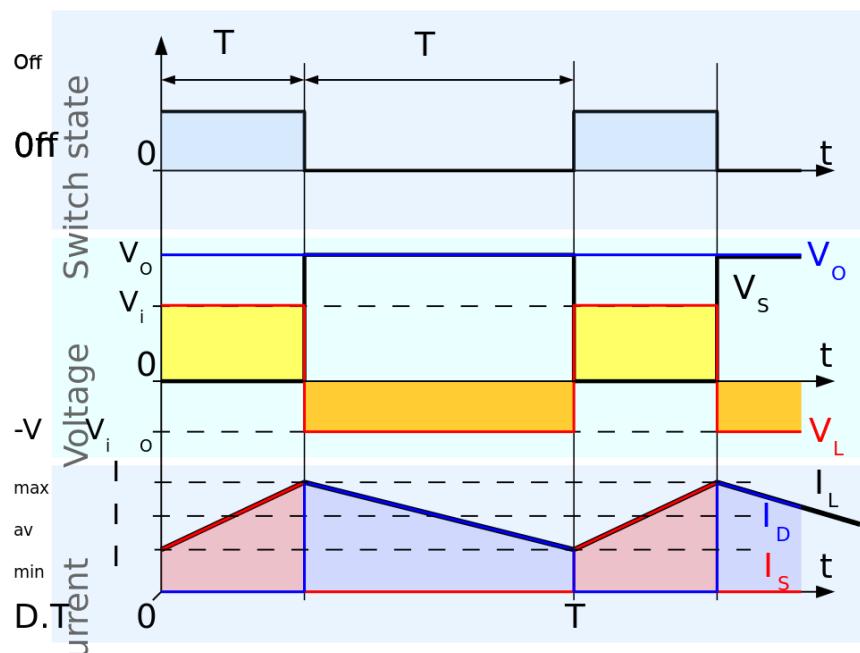


Figura 4.3: Formas de onda de corriente y voltaje del *Boost* en modo continuo. [20]

En la anterior figura podemos ver que la corriente en la bobina se incrementa linealmente cuando el interruptor está en ON y disminuye cuando el interruptor está en off pero no llega a hacerse cero nunca. Por otro lado, la tensión en la bobina oscila entre el valor de la tensión de entrada cuando el interruptor está en ON y un valor de tensión negativo, ya que cuando el interruptor está en OFF la energía almacenada en forma de campo magnético en el inductor hace que la corriente siga fluyendo por el diodo y para que esto ocurra,  $V_s > V_{in}$ , por eso la tensión en la bobina es negativa y, si  $V_s > V_{in}$ , entonces  $V_{out} > V_{in} = V_s + |V_L|$ .

Todas estas expresiones que hemos estado comentando se puede expresar matemáticamente como vamos a ver a continuación:

Si analizamos el circuito en sus dos estados de la señal de control:

$$0 < t < DT \rightarrow S \text{ ON} \& D \text{ OFF} \rightarrow V_i = L \frac{di_L}{dt} \rightarrow (\Delta i_L)_1 = \frac{V_i}{L} DT$$

$$DT < t < T \rightarrow S \text{ OFF} \& D \text{ ON} \rightarrow V_i - V_o = L \frac{di_L}{dt} \rightarrow (\Delta i_L)_2 = \frac{V_i - V_o}{L} (1 - D)T$$

Si la variación de corriente en la bobina es cero entonces:

$$(\Delta i_L)_1 + (\Delta i_L)_2 = 0 \rightarrow V_i D + (V_i - V_o)(1 - D) = 0 \rightarrow V_o = \frac{V_i}{1 - D} \quad (\text{eq 4.1})$$

Y como  $D \in [0, 1]$  entonces  $V_i > V_o$ , con lo que podemos ajustar la tensión de salida variando solamente el ciclo de trabajo y dejando fija la frecuencia.

Por otro lado, para calcular los valores de la bobina y el condensador, podemos hacer uso de la expresión del rizado de la corriente de entrada y de la tensión en el condensador:

$$S \text{ ON} \rightarrow \Delta i_i = \frac{V_i}{L} DT \rightarrow L > \frac{V_i DT}{(\Delta i_L)_{\max}}$$

$$S \text{ ON} \rightarrow \Delta V_o = \frac{I_o}{C} DT \rightarrow C > \frac{I_o DT}{(\Delta V_o)_{\max}}$$

Por último vamos a obtener la expresión de la corriente a la salida del convertidor:

$$0 < t < DT \rightarrow S \text{ ON} \& D \text{ OFF} \rightarrow i_o(t) = i_d(t) = 0$$

$$DT < t < T \rightarrow S \text{ OFF} \& D \text{ ON} \rightarrow i_o(t) = i_d(t) = I_i$$

O lo que es lo mismo:

$$i_o(t) = \begin{cases} 0 & 0 > t > DT \\ I_i & DT < t < T \end{cases}$$

Entonces:

$$I_o = I_i (1 - D) \quad (\text{eq 4.2})$$

Por último, también podemos obtener la expresión de la potencia a la salida del convertidor:

$$P_o = V_o I_o = \frac{V_i}{1 - D} I_i (1 - D) = V_i I_i \rightarrow P_o = P_i \quad (\text{eq 4.3})$$

Es decir, si no hay pérdidas, la potencia entregada a la carga es igual a la potencia de entrada con lo que tendríamos una **eficiencia del 100%** en nuestro convertidor. Y esa la propiedad que más nos interesa para nuestra aplicación, ya que teóricamente podríamos obtener el punto de máxima potencia sin perder potencia en este dispositivo.

#### 4.2.2 Modo de conducción discontinuo

El modo discontinuo se da cuando la cantidad de energía requerida por la carga es suficientemente pequeña como para poder ser transferida en un tiempo menor que el periodo de conmutación. La única diferencia es que en este caso la corriente a través del inductor cae hasta cero durante parte del periodo de conmutación. En la figura más abajo podemos ver las señales en el elevador funcionando en modo discontinuo.

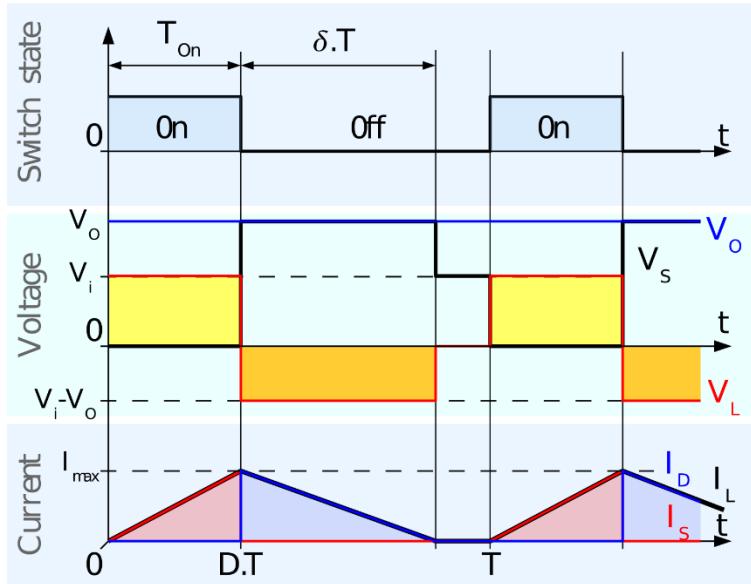


Figura 4.4: Formas de onda de corriente y voltaje del *Boost* en modo discontinuo. [20]

Como podemos ver las formas de onda en modo discontinuo son similares a las que se obtienen cuando el dispositivo opera en modo continuo, la diferencia radica en que la corriente en la bobina se hace cero durante parte del periodo de conmutación. Para entenderlo mejor vamos a analizar matemáticamente lo que ocurre:

$$0 < t < DT \rightarrow S \text{ ON} \& D \text{ OFF} \rightarrow V_i = L \frac{di_L}{dt} \rightarrow (\Delta i_L)_1 = \frac{V_i}{L} DT$$

$$DT < t < (\Delta + D)T \rightarrow S \text{ OFF} \& D \text{ ON} \rightarrow V_i - V_o = L \frac{di_L}{dt} \rightarrow (\Delta i_L)_2 = \frac{V_i - V_o}{L} \Delta T$$

Si la variación de corriente en la bobina es cero entonces:

$$\begin{aligned} (\Delta i_L)_1 + (\Delta i_L)_2 &= 0 \rightarrow \frac{V_i}{L} DT + \frac{V_i - V_o}{L} \Delta T = 0 \rightarrow V_o = V_i \frac{V_i (\Delta + D)}{\Delta} \\ \rightarrow \Delta &= \frac{V_i}{V_o - V_i} D \quad (\text{eq 4.4}) \end{aligned}$$

$(D + \Delta)$  es el instante en el que el valor de la corriente se anula. En este caso necesitaremos otra expresión que nos relacione  $\Delta$  con el valor de la corriente media a la salida del convertidor, la cual la podemos obtener de la corriente media a la salida:

$$I_o = \frac{1}{T} \frac{1}{2} \Delta T i_{Lmax} = \frac{1}{2} \Delta i_{Lmax}$$

Y como también conocemos el valor máximo de la corriente en la bobina:

$$i_{Lmax} = \frac{V_i}{L} DT$$

Entonces la corriente media en la bobina se puede expresar de la siguiente manera:

$$I_o = \frac{1}{2} \Delta i_{Lmax} = \frac{1}{2} \frac{V_i}{V_o - V_i} D \frac{V_i}{L} DT \rightarrow I_o = \frac{V_i^2 D^2 T}{2L(V_o - V_i)} \quad (eq\ 4.5)$$

Y por tanto la ganancia de tensión podemos expresarla como:

$$\frac{V_o}{V_i} = 1 + \frac{V_i D^2 T}{2L I_o} \quad (eq\ 4.6)$$

Si comparamos la expresión 4.6 con la ganancia en tensión del dispositivo en modo de conducción continua (ecuación 4.1), vemos que es mucho más compleja. Además, en modo continuo la ganancia solo depende del factor activo D, mientras que en el modo discontinuo también depende del valor de la inductor L, la frecuencia de conmutación y la corriente de entrada. Por ello vamos a tratar de operar siempre en el modo de conducción continua, por lo que fijaremos una frecuencia de conmutación lo suficientemente alta que nos garantice que operamos en modo de conducción continua, o lo que es lo mismo, usaremos un periodo de conmutación lo suficientemente pequeño para que la corriente en la bobina no se anule y, de este modo, el dispositivo no pase a modo de conducción discontinuo y también deberemos tener en cuenta la potencia de salida que marca el régimen de operación.

#### 4.2.3 Límite de operación

Para operar en el modo de conducción continua podemos calcular la potencia límite para la que se produce el cambio de modo continuo a discontinuo:

$$I_o = \frac{1}{T} \frac{1}{2} (1 - D) T i_{Lmax} = \frac{1}{2} (1 - D) \frac{V_i}{L} DT \rightarrow \frac{V_o}{R} = \frac{1}{2} (1 - D) \frac{(1 - D)V_o}{L} DT \rightarrow \\ R \leq \frac{2L}{(1 - D)^2 DT} \quad (eq\ 4.7)$$

Una vez que ya conocemos las expresiones que caracterizan este dispositivo, nos toca diseñarlo.

### 4.3 Diseño del convertidor DC-DC elevador o *Boost*

Para obtener un diseño del elevador fijaremos los valores de algunos componentes y comprobaremos que con ellos operamos en régimen continuo. Los valores del inductor y el capacitor que utilizaremos serán:  $L = 1 \text{ mH}$  y  $C = 220 \mu\text{F}$ , y la frecuencia de trabajo será:  $f_c = 30 \text{ kHz}$ , y ahora vamos a comprobar cuál es la potencia mínima que debe solicitar la carga para funcionar en modo continuo o, en otras palabras, el valor máximo de la resistencia de carga. Para ello suponemos el peor caso del ciclo de trabajo, (un ciclo de trabajo próximo a 0),  $D = 0.1$  por ejemplo y, haciendo uso de la ecuación 4.7, entonces:  $R < 740 \Omega$ . A continuación vamos a simular el dispositivo utilizando una carga de  $220 \Omega$ . Como interruptor utilizaremos un IRF 540N, que es un MOSFET de canal N, el diodo será un IN4004 UF. Este diodo debe ser UF (Ultra Fast) para que se transmita toda la energía de la bobina al condensador en un periodo de conmutación.

### 4.4 Simulación del convertidor elevador

En este apartado vamos a realizar la simulación del elevador con PSPICE:

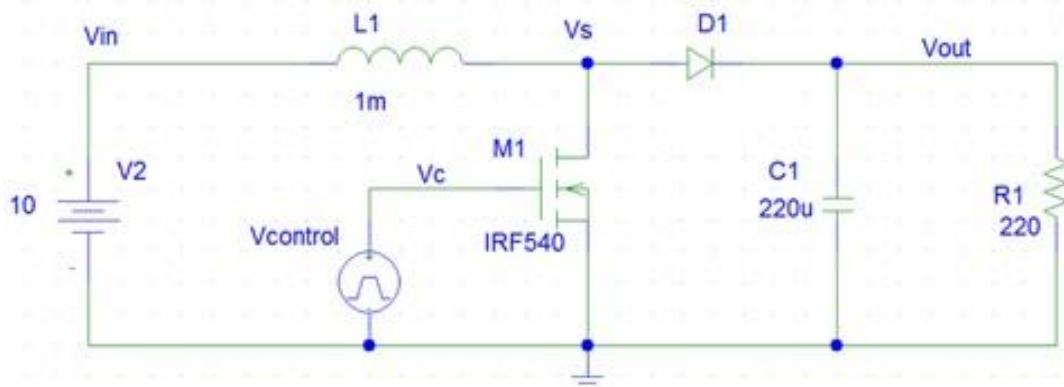


Figura 4.5: Elevador *boost* diseñado.

Para la simulación de esta etapa alimentaremos con una fuente de continua y un generador de señales cuadradas con una frecuencia de 30 kHz y un ciclo de trabajo del 70 %. En la figura siguiente podemos ver los resultados de la simulación.

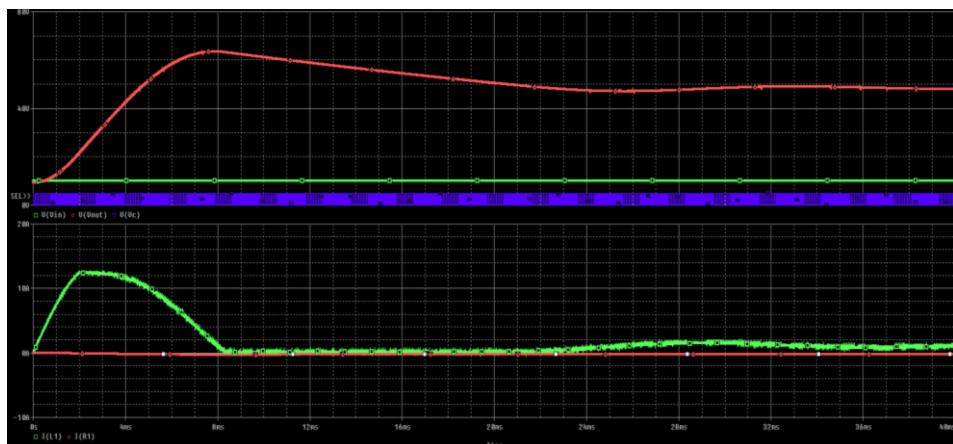


Figura 4.6: Voltajes de salida y entrada obtenidos en la simulación del convertidor *Boost*.

Como vemos en la gráfica superior de la figura 4.6, para una señal de entrada de 10 V con ciclo de trabajo del 70%, tenemos una señal de salida de unos 50 V. Por otro lado, en la gráfica inferior vemos que la corriente de salida es algo inferior a la corriente de entrada. Por eso nuestro objetivo será ajustar el ciclo de trabajo para hallar el valor que maximice el producto de la tensión y la corriente. El pico de corriente se debe al estado transitorio del sistema hasta que se estabiliza.

#### 4.5 Implementación física del convertidor *Boost*

Una vez hemos comprobado que nuestro diseño es válido, pasamos a implementarlo físicamente. En la última figura del capítulo podemos ver la situación del convertidor en la placa del proyecto. Para comprobar que funciona correctamente conectamos una fuente de continua a la entrada y un generador de señales con una señal de 30 kHz y un ciclo de trabajo del 50%, para controlar la conmutación del interruptor. Más adelante veremos cómo generar la señal de control con Arduino y como variar el ciclo de trabajo de manera automática. En la figura de abajo comprobamos que el circuito funciona correctamente.

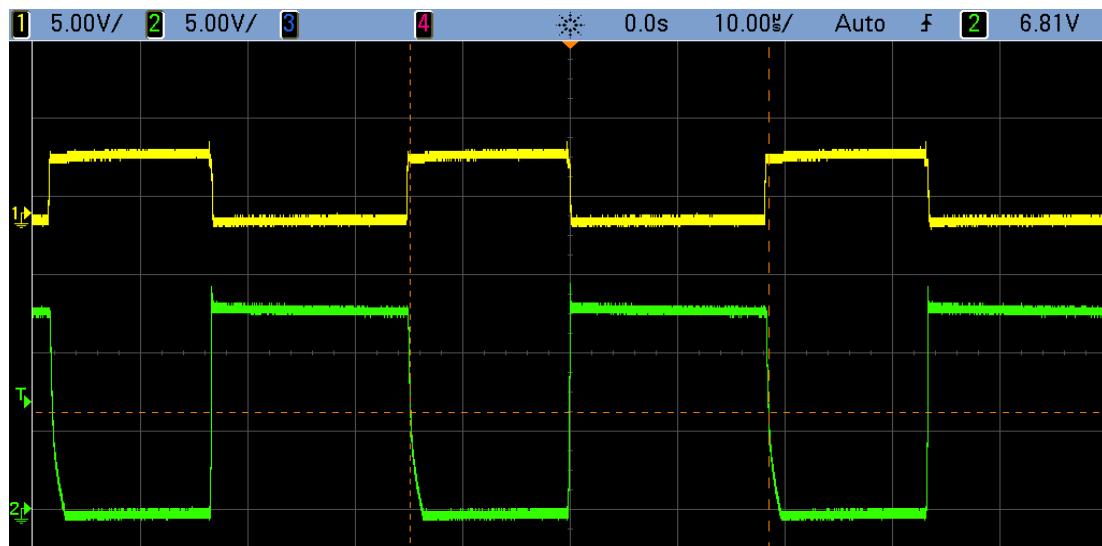


Figura 4.7: Señal de control y tensión en el drenador

En la parte superior de la figura anterior podemos ver la señal de control en la puerta del transistor y abajo tenemos la tensión en el drenador. Como vemos la tensión en el drenador es de unos 15 V y a la salida obtuvimos cerca de 20 V, todo ello con un tensión de entrada de 5 V. Por lo que podemos afirmar que el elevador está funcionando correctamente.

#### 4.6 Generación de la señal de control con Arduino

En este punto vamos a explicar cómo programar Arduino para generar la señal de control que conmutará el convertidor elevador. Para ello, tenemos que tener claro que queremos generar una señal de control de un frecuencia  $f_c = 30 \text{ kHz}$ , con un ciclo de trabajo variable que nos permita obtener el punto de máxima potencia, para lo que necesitaremos realizar medidas de la tensión y la corriente, pero eso lo veremos más adelante.

Para generar esta señal de control utilizaremos el timer/Counter1 de 16 bits de Arduino en el modo Clear Time on Compare Match (CTC Mode). En este modo se utiliza el registro ICR1 para inicializar a cero el contador y el registro OCR1A para manipular el tiempo que la señal está en alta. En la figura siguiente podemos ver in diagrama temporal del contador en este modo de operación:

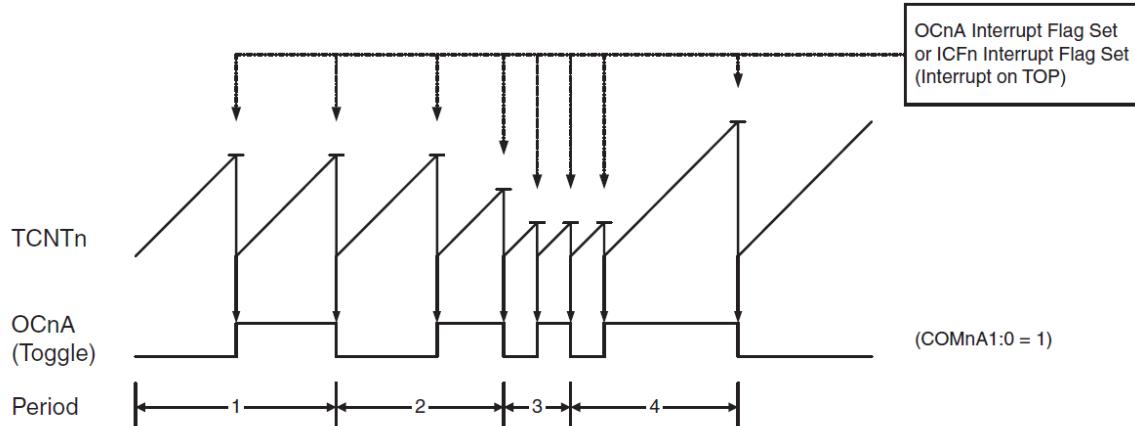


Figura 4.8: Diagrama temporal del temporizador 1 de Arduino en el modo CTC. [21]

Para hallar el valor que deben tener estos registros hacemos uso de la expresión de la frecuencia para este modo:

$$f_{OCnA} = \frac{f_{clk\_I/O}}{2N(1 + ICR1)} \quad (eq\ 4.8)$$

Donde:

- $f_{clk\_I/O}$ = Frecuencia de reloj de Arduino (16 MHz).
- $f_{OCnA}$ : Frecuencia deseada (30 kHz).
- N = Prescaler, el cual nos permite obtener un amplio rango de frecuencias. En nuestro caso usaremos N = 8.

En este modo de operación  $ICR1$  sería el valor top del contador, que al tratarse de un contador de 16 bits podría tener cualquier valor en el rango  $[0, 2^{16}]$ . Si sustituimos los valores en la expresión obtenemos que el valor de top será:

$$ICR1 = \frac{f_{clk\_I/O}}{2Nf_{OCnA}} - 1 = \frac{16 \cdot 10^6}{2 \cdot 8 \cdot 30 \cdot 10^3} - 1 = 33$$

$ICR1 = 33$ , y el tiempo en alta lo marcaremos con el registro  $OCRnA$ , cuya salida está en el Pin 9. Por ejemplo,  $OCR1A = 16$  equivaldría a un ciclo de trabajo del 50% aproximadamente.

## 4.7 Acondicionamiento

En este apartado veremos los sistemas que hemos utilizado para el acondicionamiento de las señales de esta etapa.

### 4.7.1 Driver de potencia

El transistor IRF540N requiere un driver de potencia, ya que la corriente que demanda este componente puede ser demasiado elevada para Arduino, haciendo que se queme el pin al que está conectada la puerta del transistor. El driver utilizado es el TC4427 que puede suministrar una corriente de 1.5 A y tiene un tiempo de propagación de 40 ns, lo suficientemente rápido como para no afectar a la conmutación ya que, si recordamos, estamos trabajamos a una frecuencia de 30 kHz (33,3  $\mu$ s) en el caso del convertidor *Boost* y, como ya veremos más adelante, a 50 kHz (20  $\mu$ s) en el caso del inversor. En la figura 4.9 podemos ver la configuración de los pines de este componente.

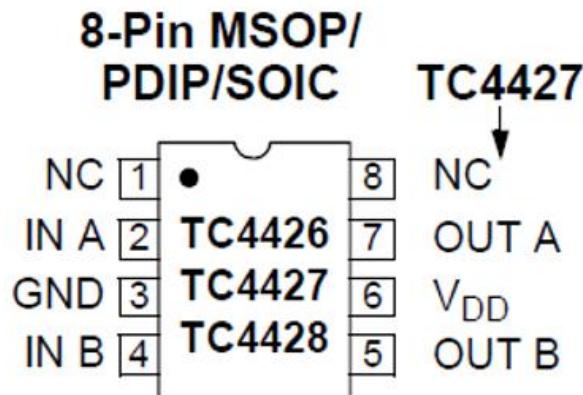


Figura 4.9: Pines del driver de potencia del MOSFET. [22]

Como vemos este driver de potencia tiene dos canales lo que nos será muy útil ya que el inversor, sobre el que hablaremos en el siguiente capítulo, también requiere drivers de potencia para los transistores.

### 4.7.2 Limitación de la tensión de salida del convertidor

Un aspecto muy importante que debemos tener en cuenta es la etapa a la que conectaremos el convertidor elevador, que en nuestro caso será un inversor, el cual estará formado por transistores de canal N y P que soportan tensiones de entre 55 V y 100 V y corrientes de varios amperios. Por ello, limitaremos la tensión de salida del convertidor *Boost* con un diodo zéner de 50 V que evite que la tensión de salida supere este valor y así no romperemos los transistores. Y en cuanto a la corriente, no debería preocuparnos porque la placa solar que utilizaremos no superará los 100 mA a plena exposición. Aunque utilizaremos varias de estas placas los transistores y demás componentes no se verían afectados. En la figura 4.10 podemos ver dónde se coloca este diodo:

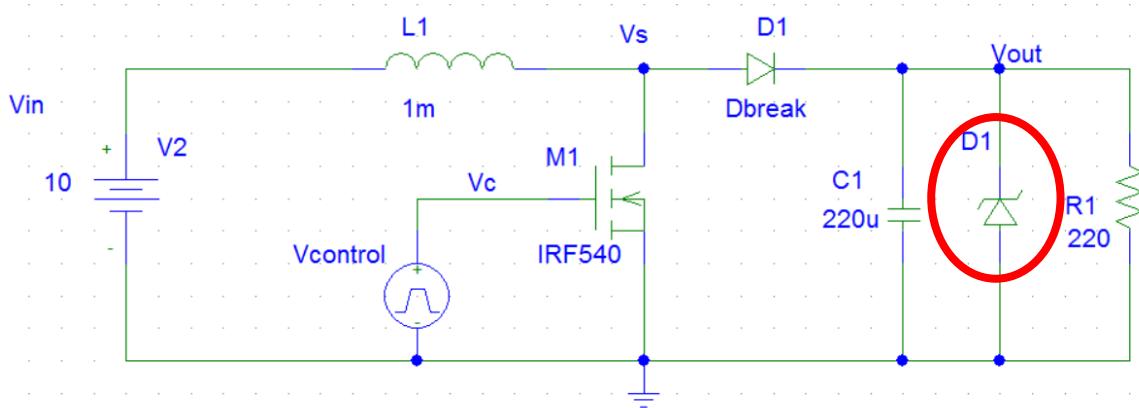


Figura 4.10: Diodo zéner para limitar la tensión de salida del *Boost*.

Cuando el diodo Zéner se polariza inversamente sólo dejará pasar una tensión constante. Actúa como un limitador de la tensión de salida.

## 4.8 Sensores

En este apartado veremos cómo diseñar los sensores empleados para medir la tensión y la corriente de entrada.

### 4.8.1 Sensor de tensión

Para medir la tensión de entrada tenemos que saber que tensión puede proporcionar el panel fotovoltaico como máximo. En nuestro caso, el panel puede suministrar hasta unos 10 V. Como la tensión de entrada máxima que pueden soportar los pines de Arduino es 5 V, entonces tenemos que diseñar un divisor de tensión que divida la tensión de entrada por dos. Un aspecto importante en este punto es que los valores de los resistores que utilicemos no tienen que ser exactos, ya que podremos ajustar las medidas que hagamos de la tensión mediante el firmware cargado en Arduino. Para hallar la relación entre el valor medido en la entrada del ADC del microcontrolador y el valor real, realizaremos varias lecturas con el polímetro y calcularemos la relación proporcional que hay entre ellas para diferentes ciclos de trabajo, esto es, una vez que tengamos el convertidor funcionando, mediremos con un polímetro la tensión de entrada real y la compararemos con la medida que nos da Arduino y multiplicaremos este valor por un factor de escala de manera que el valor obtenido sea igual al valor real.

#### 4.8.1.1 Divisor de tensión

El divisor de tensión es una configuración de circuito eléctrico que reparte la tensión de una fuente entre una o más impedancias conectadas en serie. La configuración y algunas características de este circuito las podemos ver en la figura 4.11.

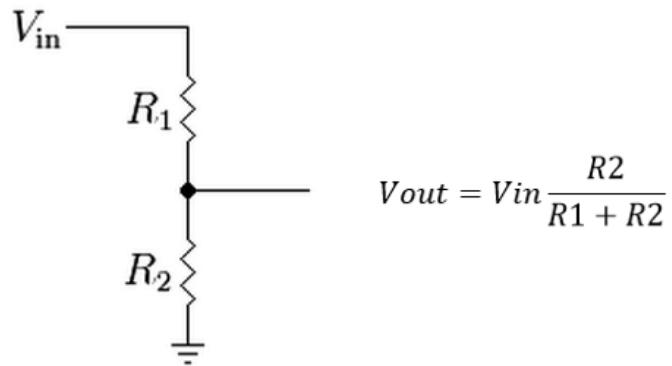


Figura 4.11: Divisor de tensión. [23]

Además de un divisor de tensión, también necesitamos un seguidor de tensión conectado a  $V_{out}$ , ya que en caso contrario el valor de esta tensión podría variar si conectamos la salida directamente a Arduino, puesto que podría fluir corriente por el pin de Arduino.

#### 4.8.1.2 Seguidor de tensión

El seguidor de tensión es un circuito que proporciona a la salida la misma tensión que a la entrada, independientemente de la carga que se le acopla, o lo que es lo mismo, independientemente de la intensidad de corriente que se le demande. En la figura siguiente, podemos ver el esquemático de este circuito

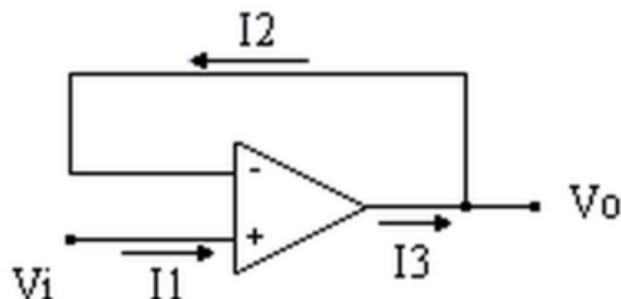


Figura 4.12: Seguidor de tensión. [24]

Como podemos ver, el seguidor de tensión no es más que un amplificador operacional en configuración inversora con ganancia 1. Este dispositivo tiene una impedancia de entrada muy elevada por lo que no fluirá corriente por la salida del divisor de tensión y no se modificará la medida (Para ver el esquema final y dónde se coloca el sensor de tensión ver Figura 4.15). A continuación vemos el amplificador operacional que utilizaremos para implementar este módulo.

#### 4.8.1.3 Amplificador operacional LM358

El integrado que usaremos para el diseño del amplificador será el LM358, que posee dos amplificadores operacionales de bajo consumo y alta ganancia que están diseñados para operar desde una sola fuente de potencia sobre un amplio rango de voltajes. En la figura siguiente podemos ver la configuración de los pines de este integrado.

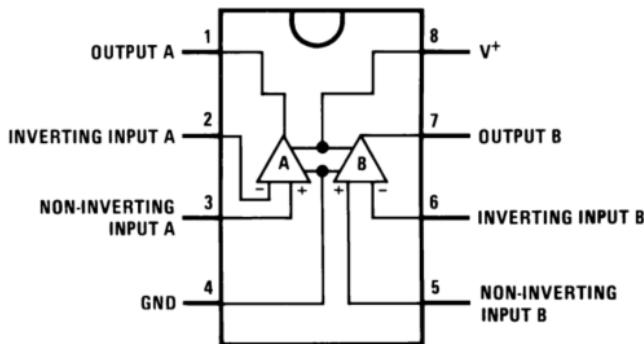


Figura 4.13 Configuración de los pines del circuito integrado LM358. [25]

Como veremos más adelante necesitaremos varios de estos operacionales para implementar los sensores que necesitamos.

#### 4.8.2 Sensor de corriente

Para medir la corriente de entrada vamos utilizar una resistencia de  $1\ \Omega$  que colocaremos entre tierra y el ánodo de la placa solar, de esta manera la corriente que circule por la placa será igual a la tensión del ánodo según la Ley de Ohm. (Ver figura 4.15):

$$I = \frac{V}{R} \quad \text{Si } R = 1\Omega, \text{ entonces } I = V$$

En este caso volvemos a tener el problema de que si colocamos un divisor de tensión y medimos directamente con Arduino, podemos variar el valor de la medida, y en este caso más aún, puesto que las medidas serán mucho más pequeñas, ya que estamos midiendo corriente (mA). Por ello tenemos que colocar un seguidor de tensión y, adicionalmente, vamos a colocar un amplificador que nos incremente la tensión al rango [0, 5]. En la figura siguiente podemos ver el esquema de un amplificador operacional en configuración inversora.

##### 4.8.2.1 Amplificador operacional inversor

A continuación vamos a ver cómo se diseña este dispositivo para obtener una determinada ganancia.

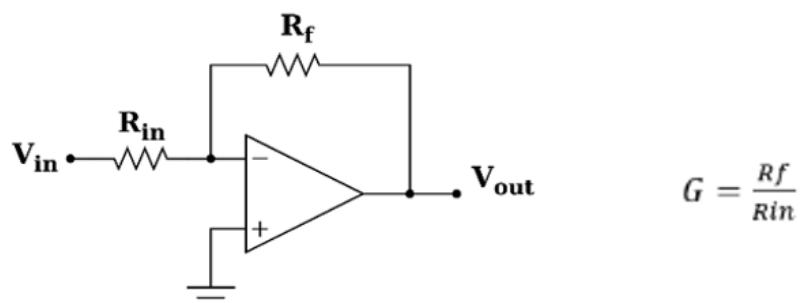


Figura 4.14: Amplificador operacional inversor. [26]

Recordamos que es posible ver el esquemático del sensor en la figura 4.15. Si ya hemos visto cómo se conectan todos los módulos podemos hablar de cómo hallar el valor de la ganancia para que las medidas sean de un valor apropiado. La tensión  $V_{sc}$  que medimos se calcula de la siguiente manera:

$$V_{sc} = I_{in} \cdot R_{sc} \cdot \frac{R_f}{R_{in}} \leq 5 V$$

Donde  $R_{sc}$  es  $1 \Omega$ . Y si la corriente máxima que puede proporcionar la placa es:  $I_{max} = 100 \text{ mA}$ , entonces:

$$100mV \cdot \frac{R_f}{R_{in}} \leq 5 V \rightarrow \frac{R_f}{R_{in}} \leq 50$$

Nosotros hemos utilizado unos resistor  $R_f = 100 \text{ k}\Omega$  y  $R_{in} = 2.2 \text{ k}\Omega$ . Como ya hemos comentado estos valores no tiene que ser exactos porque después calibraremos las medidas con un factor de escala cuyo valor obtendremos después de realizar varias medidas. En la figura siguiente podemos ver la configuración final del DC-DC elevador con los sensores de tensión y corriente.

#### 4.9 Simulación del *Boost* con los sensores

En este apartado podemos ver la configuración final del convertidor elevador con los sensores y los resultados de la simulación.

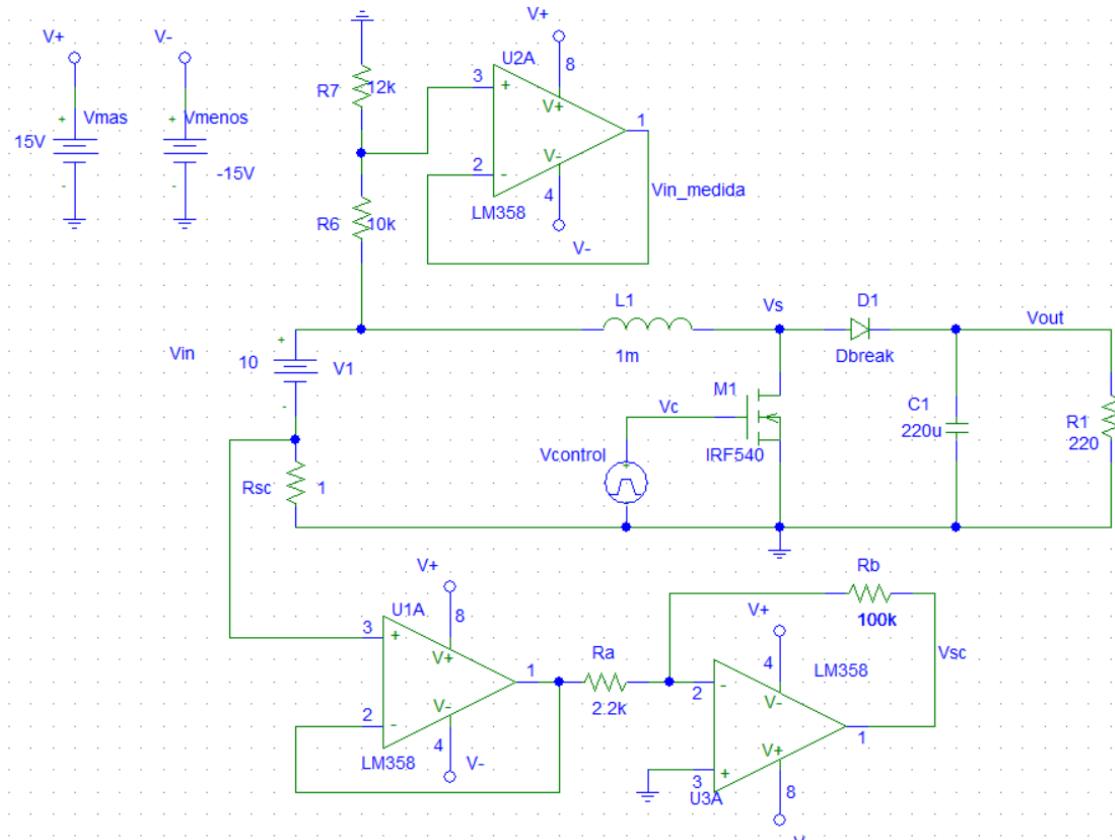


Figura 4.15: Convertidor DC-DC elevador con los sensores de  $V_{in}$  e  $I_{in}$ .

A continuación vamos a estudiar el resultado de la simulación de este circuito para cerciorarnos de si las medidas que realicemos coinciden con el valor real.

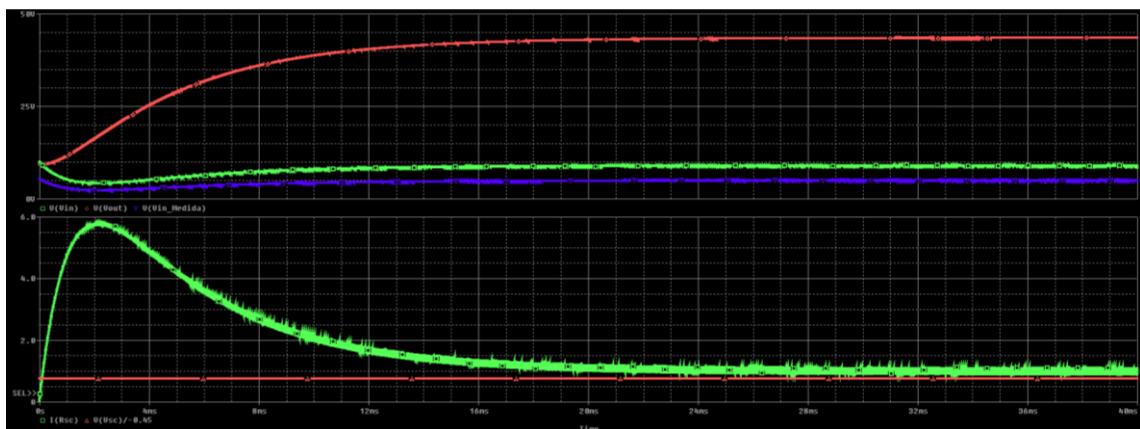


Figura 4.16. Simulación de los sensores de tensión y corriente de entrada del panel fotovoltaico.

En la gráfica superior de la figura anterior podemos ver que la tensión de entrada y la tensión media tienen un valor muy aproximado. Es a esto a lo que nos referíamos cuando hablamos sobre que el valor de las resistencias no tenía que ser exacto porque después podíamos corregir las desviaciones mediante la calibración de las medidas. Por otro lado, en la parte inferior vemos la corriente de entrada y la corriente media con el sensor que hemos diseñado. En este caso ya ha sido aplicado un factor de escala para ver un ejemplo de cómo se ajustan los valores medidos para que sean más próximos a los valores reales.

#### 4.10 Calibración de los sensores

Como ya sabemos las medidas de la tensión y la corriente de entrada que realiza Arduino a través del convertidor ADC están en el rango [0,5] V, ya que hemos diseñado los sensores de tensión y corriente para que esto ocurra así. Entonces para hallar el valor real de la tensión y la corriente de entrada necesitamos aplicar un factor de escala a las medidas realizadas. Para ello tomamos varias medidas con Arduino y medimos también el valor real con un polímetro, después representamos los resultados y hallamos la pendiente de la recta de calibración.

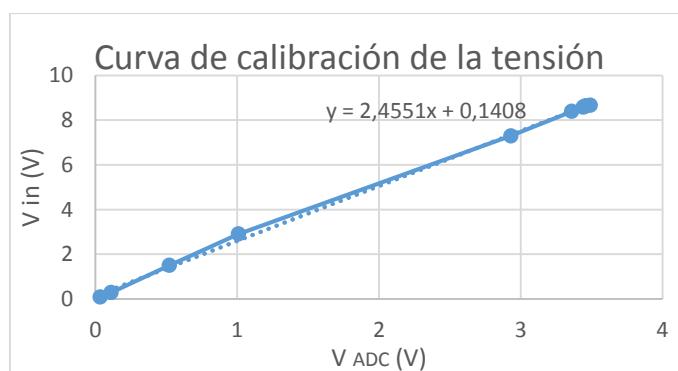


Figura 4.17: Curva de calibración de la tensión.

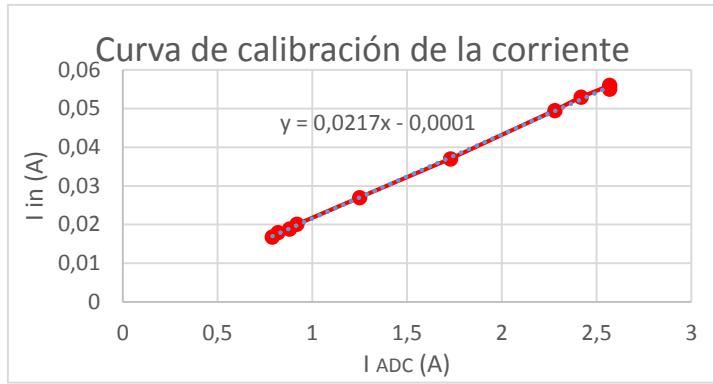


Figura 4.18: Curva de calibración de la corriente.

Como vemos en las figuras anteriores, la relación entre el valor de entrada real y el valor en la entrada del ADC de Arduino se relacionan proporcionalmente, es decir, si ajustamos los datos linealmente y calculamos la pendiente de la recta obtendremos el factor de escala que hemos de aplicar a las medidas realizadas. Los factores de escala son:

$$V_{real} = 2.455 V_{ADC}$$

$$I_{real} = 0,022 I_{ADC}$$

#### 4.11 Medición del factor de llenado (Fill Factor)

Para terminar con los aspectos técnicos de este capítulo vamos a explicar cómo medir el factor de llenado del panel solar. Recordamos que el factor de llenado es una medida de la eficiencia de un panel fotovoltaico y para medirlo necesitaremos realizar un par de medias,  $V_{oc}$  e  $I_{sc}$ . En este apartado vamos a explicar cómo medir la tensión en circuito abierto y la corriente en cortocircuito, ya que si recordamos el factor de llenado viene dado por la siguiente expresión:

$$FF = \frac{P_m}{V_{oc} \cdot I_{sc}}$$

Donde:

- $V_{oc}$  : V open circuit.
- $I_{sc}$  : I short circuit.

La manera de medir la corriente en cortocircuito es muy sencilla, solo tenemos que poner la señal de control del transistor del convertidor en alta de manera que el cátodo del panel queda conectado al inductor y este queda cortocircuitado con tierra. Una vez hayamos conseguido hacer esto con Arduino, tomamos la medida de la corriente que se mide en este estado.

Para realizar la medida del voltaje en circuito abierto, se ha introducido un nuevo transistor de potencia que conecta uno de los terminales de la carga con tierra, cuando su tensión de puerta es superior a la tensión umbral. Para dejar flotante el panel y poder medir  $V_{oc}$  basta con abrir este transistor y el del convertidor DC-DC. En la Figura 4.19 podemos ver dónde se conecta este transistor. Recordamos que los IRF 540N debe de tener un driver de potencia TC4427. Como este integrado tiene dos canales, podemos utilizar el segundo canal del que utilizamos para el transistor del convertidor elevador.

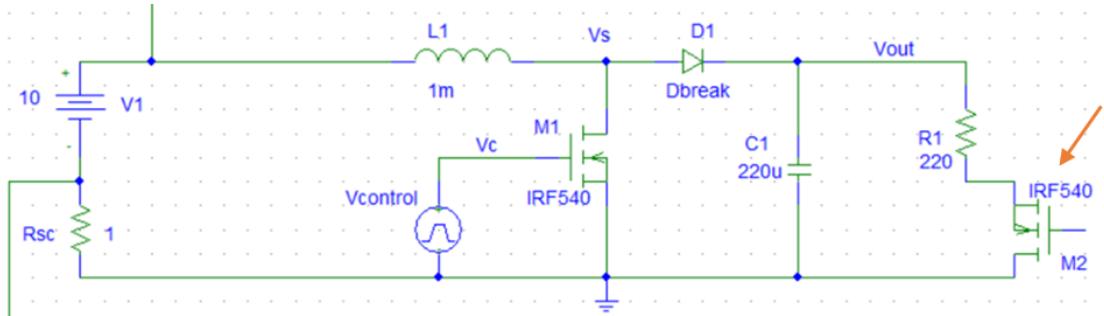


Figura 4.19: Conexión del transistor que permite por el circuito en abierto.

Para terminar este capítulo vamos a identificar dónde están situados los componentes de esta etapa en la placa del proyecto.

#### 4.12 Distribución de los componentes

En este apartado podemos identificar los componentes que forman el convertidor elevador.

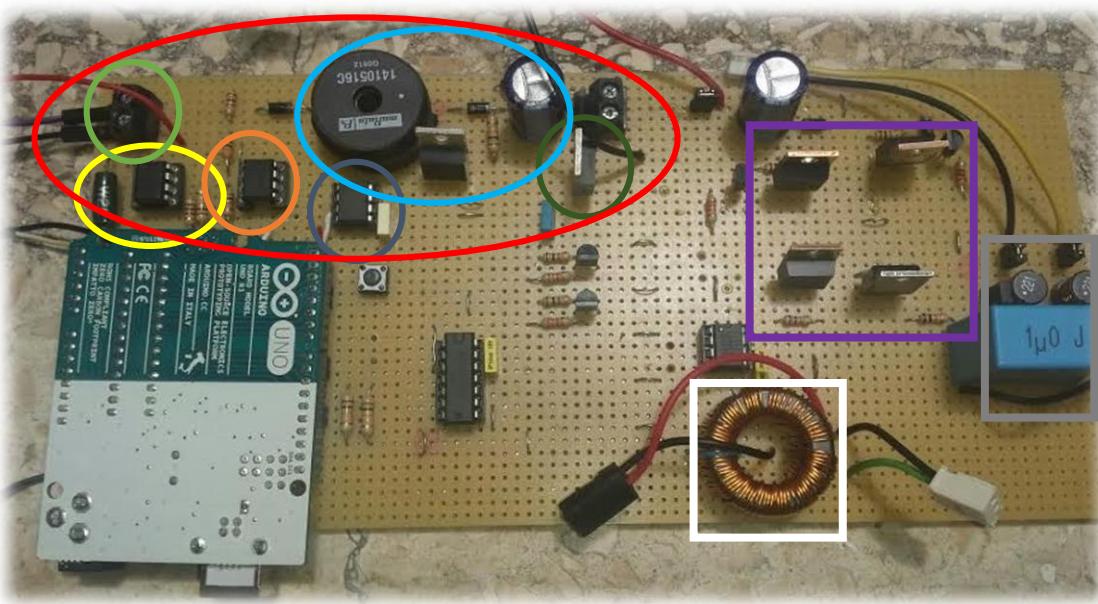


Figura 4.20: Situación de los componentes del convertidor DC-DC elevador.

En el círculo rojo podemos ver todos los componentes que conforman esta etapa. En color amarillo tenemos el sensor de corriente, con la resistencia de  $1 \Omega$  y el amplificador inversor cuya salida está conectada a Arduino. En naranja podemos ver el sensor de tensión. Cada uno de los sensores tiene sus correspondientes resistencias; para el divisor de tensión y para el amplificador inversor. En color azul oscuro vemos el driver de potencia de los transistores, que suministra potencia a los MOSFET. En verde oscuro encontramos el transistor que nos permite poner el circuito en abierto para medir la corriente  $I_{sc}$  y, en azul claro están los componentes que forman el convertidor DC-DC elevador, una bobina, un MOSFET, un diodo y un condensador y en verde claro está el conector de la placa solar. En los cuadros púrpura, gris y blanco tenemos el inversor, el filtro LC y un prototipo de transformador respectivamente.

---

## **5. Desarrollo del inversor**

En este capítulo vamos a aprender los aspectos relacionados con el diseño de un inversor, así como la utilidad del mismo, y los componentes que utilizaremos para su implementación. También veremos cómo generar las señales de control que nos permitirán conmutar el puente de transistores y las características que han de tener estas señales dependiendo del tipo de carga que conectemos a la salida del inversor.

---

## 5.1 Introducción al inversor

Un inversor es un convertidor DC-AC, es decir, su función es cambiar una señal de entrada de continua a una señal de salida alterna con la magnitud y la frecuencia requeridas. Los inversores se utilizan sobre todo en aplicaciones con motores de alterna de velocidad variable, sistemas de alimentación ininterrumpida y dispositivos de corriente alterna que funcionen a partir de fuentes de continua como baterías o paneles fotovoltaicos. Antes de que veamos la topología de un inversor, mencionar que hay muchos tipos de ellos dependiendo de la aplicación que queramos implementar.

## 5.2 Tipos de inversores

Los tipos de inversores se pueden dividir en varias categorías:

- Según la topología:
  - Monofásicos: Medio puente, puente completo y push-pull.
  - Trifásicos: Puente trifásico.
- Según el tipo de modulación:
  - Modulados: desplazamiento de fase, PWM bipolar y unipolar.
  - No modulados.

En este capítulo no vamos a explicar todas las características de cada tipo de inversor, sino que nos vamos a ceñir al dispositivo que vamos a utilizar en el proyecto, que es el inversor de puente completo. El principal motivo es que la corriente en valor medio en los bobinados del transformador es 0, lo que permite el manejo de corrientes más elevadas sin llegar a saturar el núcleo del mismo.

## 5.3 Inversor de puente completo

Para este proyecto vamos a implementar un inversor de puente completo [27], que genera una señal de salida cuadrada cuya tensión oscila con un valor de pico igual a la tensión continua de alimentación. En esta figura podemos ver la topología del inversor de puente completo.

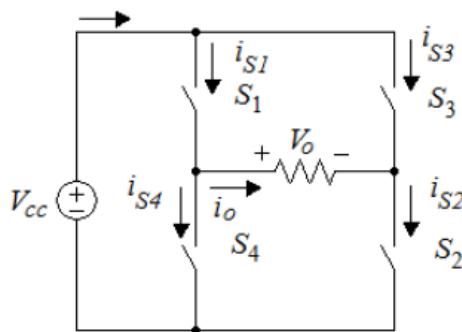


Figura 5.1: Inversor de puente completo. [28]

## 5.4 Análisis del circuito

La conmutación adecuada de los interruptores nos proporcionará la onda de salida que necesitamos. En la siguiente tabla podemos ver las posibles combinaciones de estos interruptores.

Interruptores cerrados	Voltaje de salida
S1 & S2	+Vcc
S3 & S4	-Vcc
S1 & S3	0
S2 & S4	0

Figura 5.2: Posibles combinaciones de los interruptores de un puente completo.

Cuando se conmutan los interruptores de la manera en que está indicado en la tabla de la figura anterior el circuito se comporta de las maneras que se muestran a continuación:

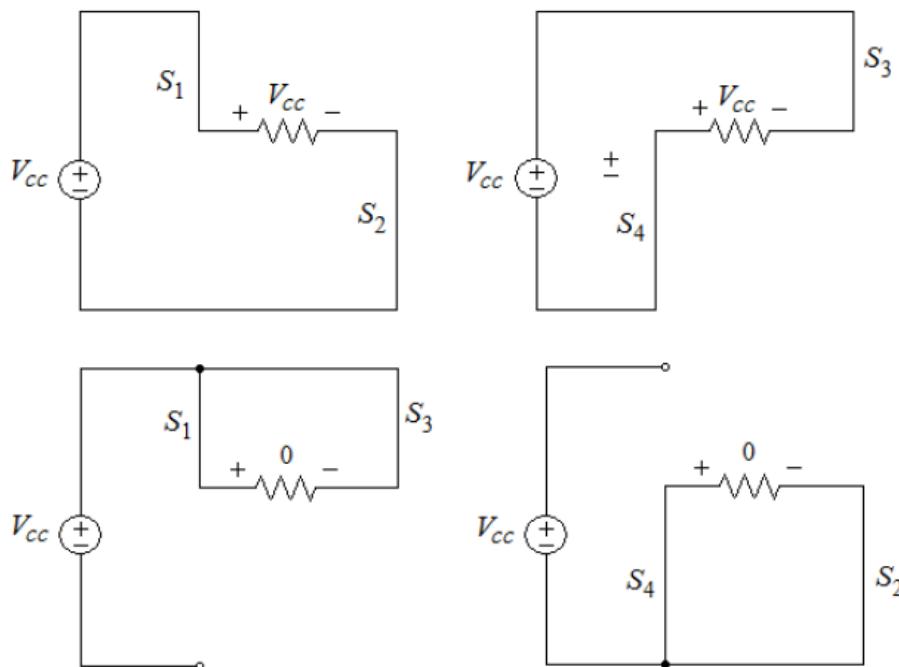


Figura 5.3: Estados del inversor de puente completo. [28]

Se puede observar que  $S_1$  y  $S_4$  no deben estar cerrados al mismo tiempo, así como  $S_2$  y  $S_3$ , de lo contrario se produciría un cortocircuito. Este es un aspecto a tener en cuenta, ya que los interruptores reales no se abren y se cierran instantáneamente, por lo que, deben tenerse en cuenta los tiempos de transición de la conmutación al generar las señales de control de los interruptores.

A nosotros nos interesa activar la pareja  $S_1$  &  $S_2$  y desactivar la pareja  $S_3$  &  $S_4$  y viceversa, de esta manera la conmutación periódica del puente produce un voltaje que oscila entre los valores  $+V_{cc}$  y  $-V_{cc}$ , generando una señal de voltaje cuadrada en la carga. Aunque esta salida alterna no es sinusoidal, puede ser una onda de alterna adecuada para algunas aplicaciones. Más adelante veremos cómo conseguir una señal sinusoidal mediante la modulación PWM y un filtrado adecuado.

Por otro lado la forma de onda de la corriente en la carga depende en sí, de la propia carga. En una carga resistiva, la forma de onda de corriente corresponde con la forma de voltaje de salida, mientras que en una carga inductiva tiene una forma distinta por ello tendremos que tener algunas precauciones a la hora de comutar el puente. Pero ahora, vamos a ver la forma de onda que se obtiene al comutar el puente de la manera en la que se ha descrito anteriormente.

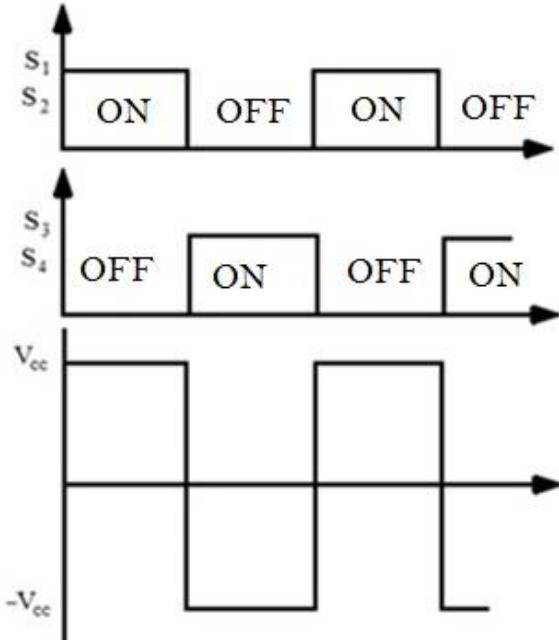


Figura 5.4: Forma de onda cuadrada obtenida a la salida del inversor. [28]

La forma de onda del voltaje de los inversores ideales debería ser sinusoidal. Sin embargo, las de los inversores prácticos no son sinusoidales y contiene ciertas armónicas. Con la disponibilidad de los dispositivos semiconductores de potencia de alta velocidad, se pueden minimizar los contenidos armónicos del voltaje de salida, o al menos reducirlos de forma importante mediante técnicas de modulación.

## 5.5 Modulación por ancho de pulsos

El proceso de modulación PWM [29] (Pulse Width Modulation) consiste en convertir una señal de referencia sinusoidal de frecuencia fundamental  $\omega_0$  en otra señal modulada por ancho de pulso. Esta transformación consiste sencillamente en comparar una referencia senoidal con una señal portadora, típicamente triangular o diente de sierra, de modo que la salida esté en nivel alto cuando la referencia sea mayor que la portadora, y en nivel bajo en caso contrario. Al ser distintas las señales de referencia y portadora en frecuencia, forma de onda y amplitud, el tiempo que pasará entre una intersección y la siguiente nunca será el mismo, de modo que los pulsos de la forma de onda de salida serán de distinta anchura, de ahí el nombre modulación de ancho de pulso. En principio, todos los esquemas de modulación pretenden crear un tren de pulsos que tenga el mismo promedio fundamental amplitud-tiempo que la señal de referencia en cualquier instante. En la figura 5.5 podemos ver un ejemplo de generación de una señal PWM.

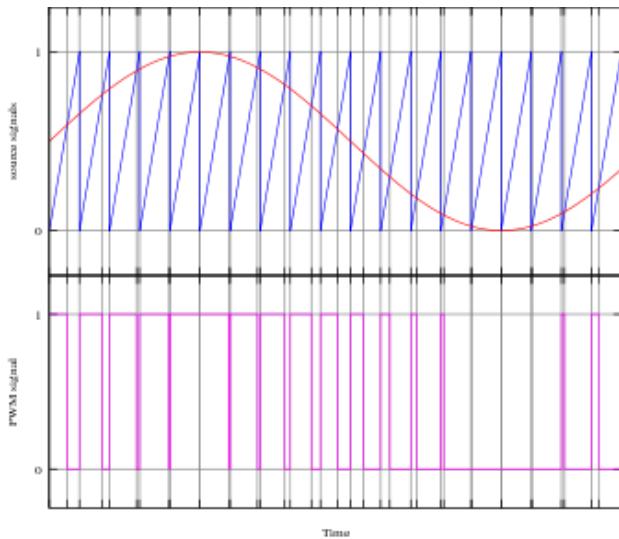


Figura 5.5: Proceso de modulación PWM. [30]

En la modulación PWM, la amplitud del voltaje de salida se puede controlar por medio de las formas de onda moduladoras. Dos ventajas de la modulación PWM son la reducción de los requerimientos de filtro para eliminar armónicos y el control de la amplitud de salida. Entre las desventajas se puede citar que los circuitos de control de los interruptores son más complejos, y que hay mayores pérdidas debido a una conmutación con mayor frecuencia. A continuación vamos a hablar sobre los tipos de modulación.

### 5.5.1 Modulación bipolar

En la versión bipolar la salida toma valores alternos entre más y menos la tensión de alimentación. En la figura siguiente se muestra el principio de la modulación por anchura de pulsos bipolar.

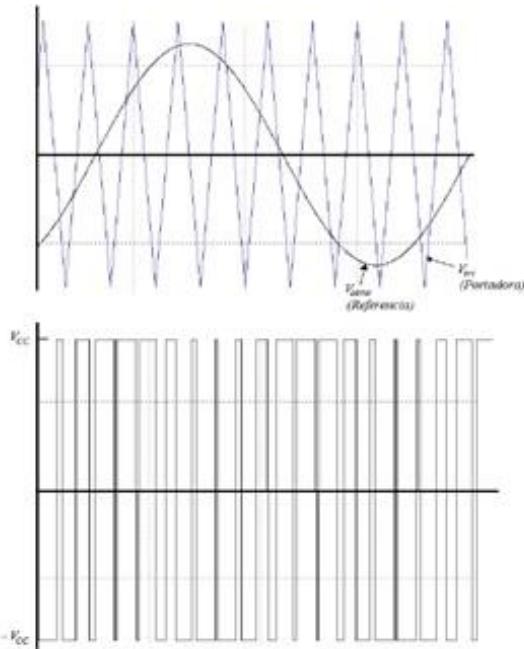


Figura 5.6: Modulación bipolar. Referencia y portadora. Salida. [28]

Cuando el valor instantáneo de la sinusode de referencia es mayor que la portadora triangular, la salida está en  $+V_{cc}$ , y cuando la referencia es menor que la portadora, la salida está en  $-V_{cc}$ , o lo que es lo mismo:

*S1 y S2 conducen cuando  $V_{seno} > V_{triángulo}$  ( $V_o = +V_{cc}$ )  
 S3 y S4 conducen cuando  $V_{seno} < V_{triángulo}$  ( $V_o = -V_{cc}$ )*

### 5.5.2 Modulación unipolar

En un esquema de modulación PWM unipolar, la salida se conmuta de nivel alto a cero, o de nivel bajo a cero, en lugar de entre niveles alto y bajo, como ocurre en la conmutación bipolar. En la Figura siguiente se muestra el principio de funcionamiento de la modulación unipolar

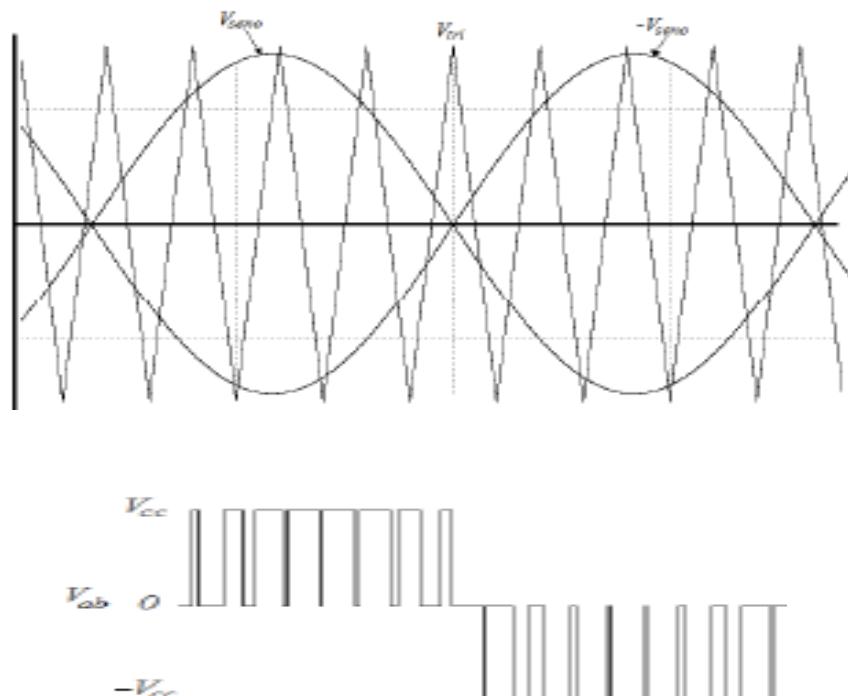
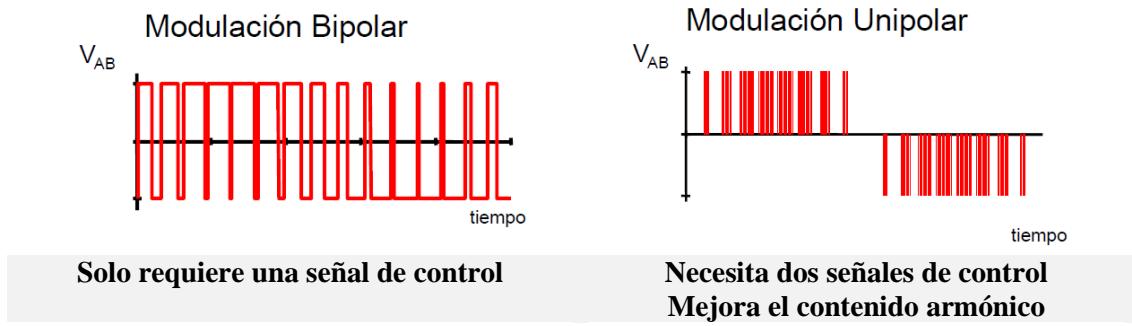


Figura 5.7: Modulación unipolar. Referencias sinusoidales y portadora triangular. Salida. [28]

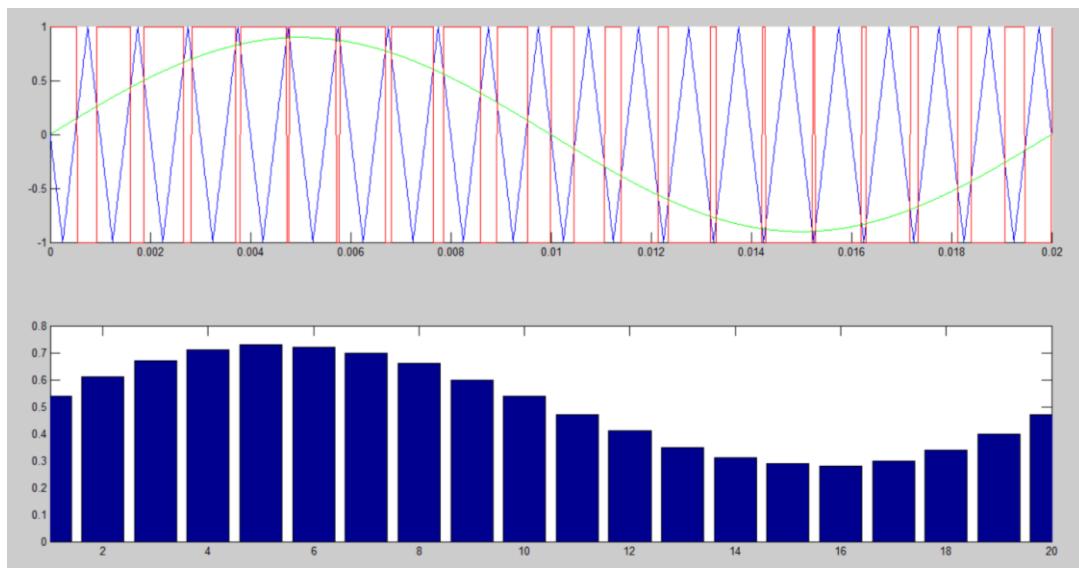
Un esquema de conmutación unipolar tiene los siguientes controles de los interruptores:

*S1 conduce cuando  $V_{seno} > V_{triángulo}$   
 S2 conduce cuando  $-V_{seno} < V_{triángulo}$   
 S3 conduce cuando  $-V_{seno} > V_{triángulo}$   
 S4 conduce cuando  $V_{seno} < V_{triángulo}$*

A continuación vamos a ver una comparativa entre ambos tipos de conmutación con el objetivo de seleccionar la más adecuada para nuestra aplicación:



Como ya comentamos en capítulos anteriores, la manera más apropiada de generar la señal de control del puente sería leyendo la frecuencia de la red eléctrica en tiempo real, y utilizar esta señal como referencia para la modulación, pero al tratarse de un proyecto orientado a docencia lo haremos de una manera más simple ya que los recursos de que dispone nuestro microcontrolador son limitados. En nuestro caso, no vamos a leer la frecuencia de la red sino que generaremos estas señales en MATLAB y calcularemos el valor de los ciclos de trabajo que necesitamos para generar nuestra señal de control, después programaremos una de las salidas PWM de Arduino para que tome los valores de ciclo de trabajo que hayamos obtenido, y con un poco de lógica digital conseguiremos las señales para cada uno de los interruptores. En la siguiente figura podemos ver los valores de los ciclos de trabajo.



Hemos de mencionar que estos valores son orientativos, y que cuando el sistema funcione podremos variar estos valores para ajustarlo cuando estemos visualizando las señales en el osciloscopio. La causa por la que las señales se desajustan es porque pasan por varias etapas que comentaremos más adelante, en las que la señal se invierte, se adelanta y se retarda, se amplifica, etc.

## 5.6 Señales de control del puente inversor

A continuación vamos a explicar cómo generar las señales de control del puente y una vez hayamos entendido como funciona la conmutación vamos a realizar una simulación con PSPICE. En primer lugar tenemos que tener en cuenta que vamos a comutar los interruptores del puente dos a dos como ya hemos comentado previamente, es decir, S1&S2 se cerraran y abrirán a la vez y S3&S4 también actuaran de manera sincronizada. En otras palabra, las señales de control de S1&S2 serán la misma, al igual que las señales de S3&S4. Lo que vamos a hacer entonces es generar una señal PWM con una frecuencia de 50 kHz (20  $\mu$ s de periodo) y una señal de 50 Hz (20 ms de periodo), esta última será nuestra señal de referencia, es decir, durante medio periodo (10 ms) el ciclo de trabajo de la señal PWM irá variando con los valores obtenidos previamente. Y durante los 10 ms restantes de la señal de referencia la señal PWM volverá a variar el ciclo de trabajo de la misma manera, después tendremos que invertir la señal cada 10 ms. En la figura siguiente podemos observar una imagen que nos ayudará a comprender lo que aquí estamos comentando.

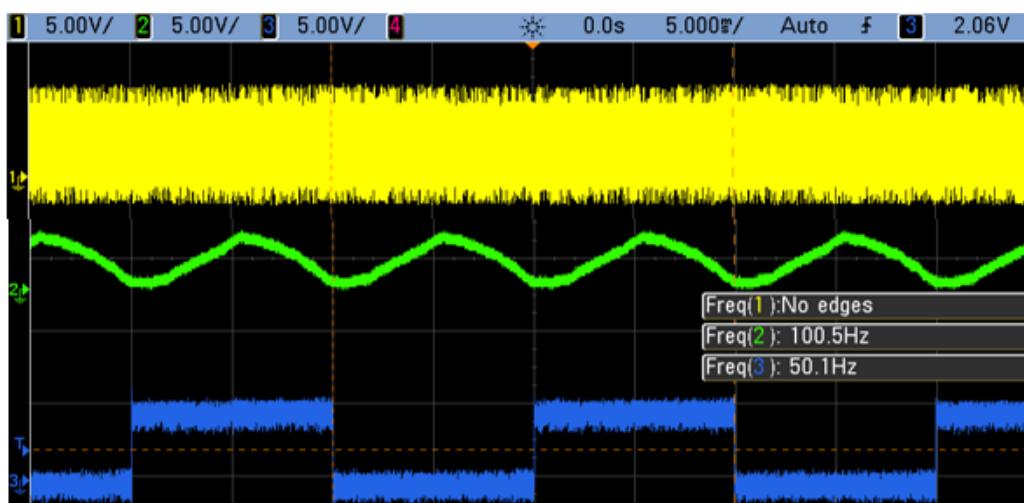


Figura 5.10: Señales PWM, PWM filtrada y 50Hz para el disparador.

La señal en amarillo de la figura anterior es una señal PWM cuyo ciclo de trabajo varía desde un valor mínimo hasta un valor máximo cada 10 ms. Después, en verde, tenemos la señal PWM tras haber sido filtrada con un paso bajo (R-C) con frecuencia de corte mucho menor que la de la señal PWM pero mayor de 50 Hz, con esto comprobamos que la PWM tiene una componente fundamental de 100 Hz (10ms), en lugar de 50 Hz (20 ms), pero esto debe de ser así por ahora. Más adelante explicaremos como invertir la señal cada 10 ms para obtener un ciclo positivo y un ciclo negativo, lo que generará una señal oscilante de 50 Hz. Pero ahora vamos a hacer zoom sobre la señal PWM para visualizar mejor lo que está ocurriendo:

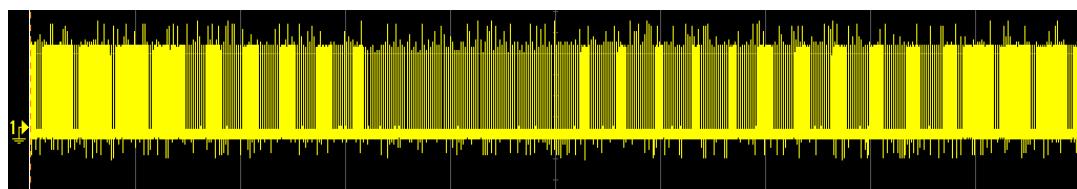


Figura 5.11: Zoom de la señal PWM.

En la figura anterior, la 5.11 se puede apreciar como la señal PWM va variando el ciclo de trabajo, pero podemos verlo mejor aún, si hacemos que el osciloscopio adquiera las muestras en el modo de alta resolución:

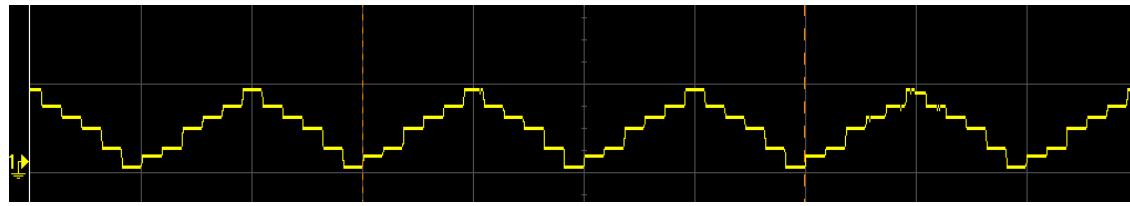


Figura 5.12: PWM visualizada en el modo de alta resolución.

### 5.6.1 Lógica digital necesaria para las señales de control

Ya que hemos entendido como funciona una señal PWM, lo que debemos hacer a continuación es mezclar la señal de 50 Hz y la de 50 kHz de manera que, por un lado, cuando la señal de 50 Hz esté en alta, la señal de 50 kHz será transmitida hacia una pareja de transistores del puente y cuando la señal de 50 Hz esté a nivel bajo, la señal de 50 kHz no será transmitida hacia esa misma pareja, por otro lado, generaremos una señal de 50 Hz negada que será mezclada con una señal de 50 kHz negada también. De esta manera conseguimos que esta señal commute la otra pareja de transistores. Para visualizar lo que estamos comentado, observemos la siguiente figura donde podemos ver la lógica que hemos implementado para mezclar las señales.

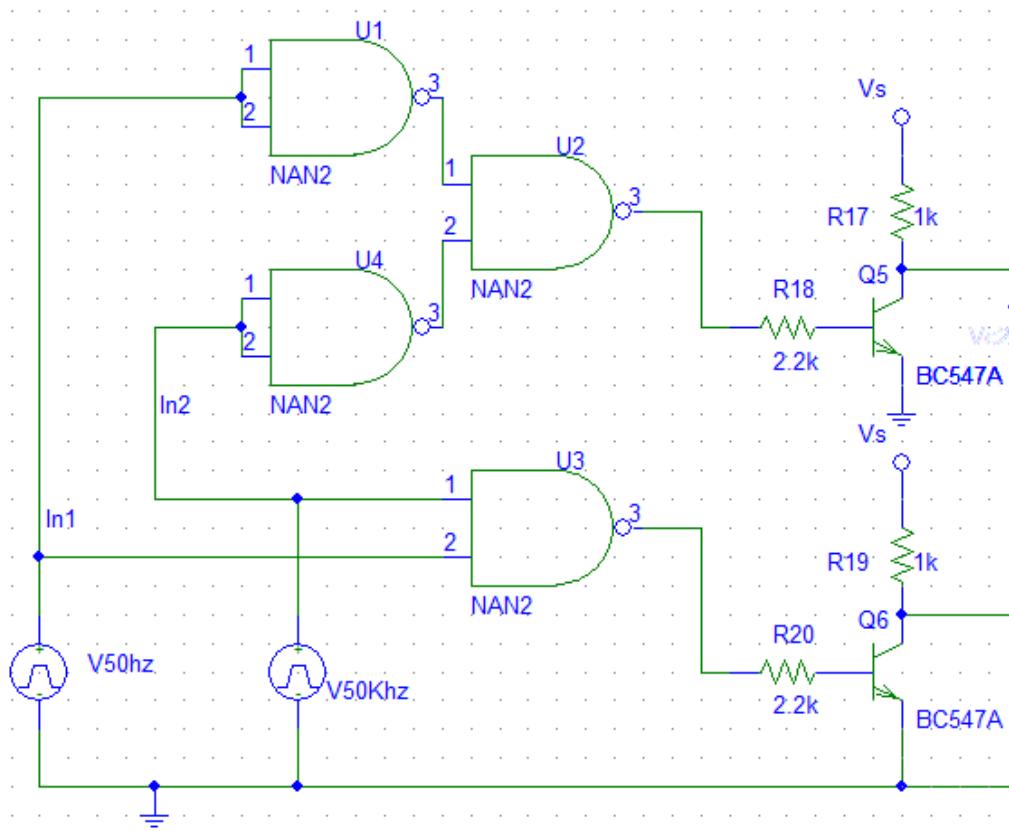


Figura 5.13: Lógica para la generación de las señales de commutación del puente.

En cuanto a la figura anterior, las puertas NAND que utilizaremos serán las del circuito integrado CD74H00, sobre el que hablaremos a continuación. En la esta figura, vemos también que hay una etapa amplificadora cuya función es aumentar la potencia que nos proporcione la salida del CD74H00. Otra manera de conseguir esto sería utilizando el driver de potencia TC4427, sobre el que ya hemos hablado, e incluso podríamos poner el driver a la salida de los bipolares, consiguiendo el mismo resultado. Más adelante, en la figura 5.16 podemos ver las señales que se obtienen a la salida de la lógica, pero antes veamos la configuración interna del circuito integrado CD74H00:

### 5.6.1 Circuito integrado CD74H00

En esta etapa solo tenemos que comentar que para implementar la lógica digital utilizaremos el circuito integrado CD74H00, cuyo diagrama funcional podemos ver en la siguiente figura.

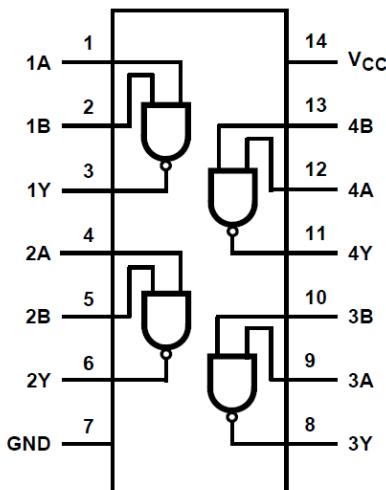


Figura 5.14: Diagrama funcional del CD74H00. [32]

Como vemos este integrado tiene cuatro puertas NAND, que son justo las que necesitamos para implementar la lógica digital que nos genera las señales de control de los interruptores

### 5.7 Simulación del puente inversor

Antes de ver la simulación de las señales que comutarán cada uno de los cuatro transistores del puente hay que comentar que nuestro puente tiene transistores N y P. Y como la señal que conmuta cada lado del puente tienen que ser iguales, entonces tendremos que añadir una etapa previa que niegue el valor de la señal de control antes de llegar el transistor tipo P. En resumen, se genera una señal de control para cada pareja, cada una de estas señales llega directamente al transistor N y pasa por una etapa previa que la niega antes de llegar al transistor P.

Otra manera de implementar el puente sería utilizar solo transistores tipo N, con lo que no tendríamos que negar las señales de control, pero necesitaríamos un driver de alta para los transistores de arriba.

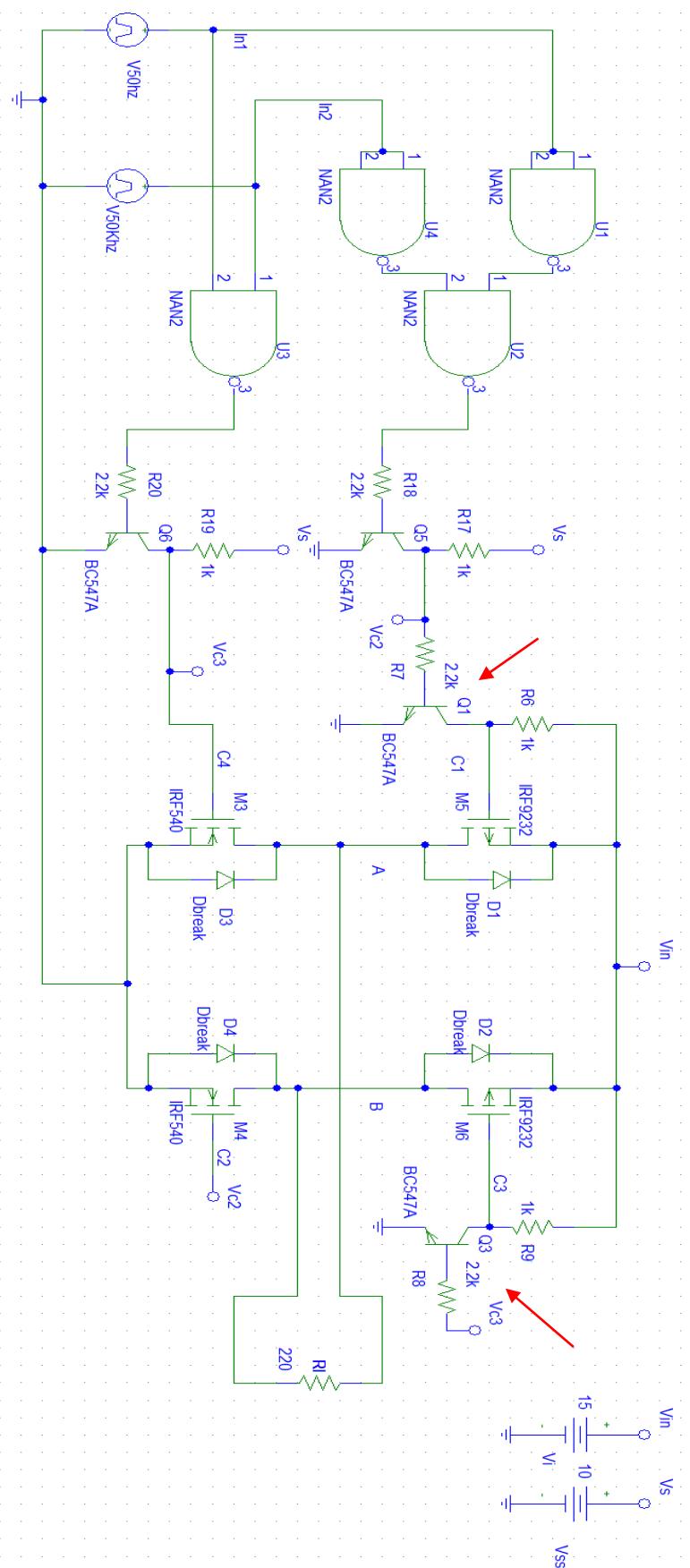


Figura 5.15: Esquemático del circuito del puente inversor.

En la anterior figura podemos ver la configuración final del puente inversor. Las flechas rojas muestran la etapa que niega la señal de control antes de llegar a los transistores de canal P. En el esquema observamos que los transistores tienen unos diodos en entiparalelo. En realidad los transistores MOSFET ya tienen estos diodos integrados pero nosotros los colocamos los 1N4004 UF por si no son lo suficientemente rápidos para la frecuencia a la que estamos trabajando. La función de estos diodos es permitir que la corriente vuelva a la fuente en el caso de utilizar una carga inductiva, es decir, en el caso de tener un transformador en la carga, el puente debe de permitir que el núcleo de la bobina se desmagnetice entre cada conmutación, para ello además debemos tener en cuenta que debe existir un tiempo de off entre un estado de la conmutación y otro, pero ya hablaremos de eso más adelante. Ahora vamos a ver las señales de control C1, C2, C3 y C4. Para situarlas en el circuito podemos fijarnos en la figura anterior, la 5.15. C3 y C4 corresponden con la señal amarilla y azul, y C1 y C2 con la verde y a roja respectivamente.

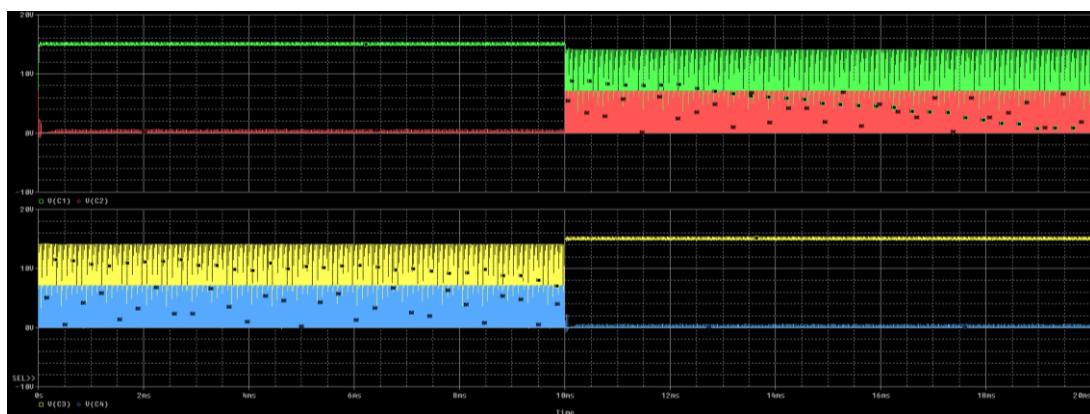


Figura 5.16: Señales necesaria para comutar el puente inversor.

Como podemos ver en la figura anterior, las señales de control de cada pareja no son exactamente iguales, sino que están invertidas, esto es porque como ya comentamos, nuestro puente tiene transistores de canal P en la parte alta y transistores canal N en la parte baja, por ello, necesitamos una etapa que nos niegue el valor de la señal de control. En la figura 5.15 que hemos visto antes podemos ver la etapa de la que estamos hablando señalada con flechas rojas.

A continuación vamos a ver qué sucede la tensión en la carga cuando el circuito está funcionando, para ello fijémonos en la siguiente figura.

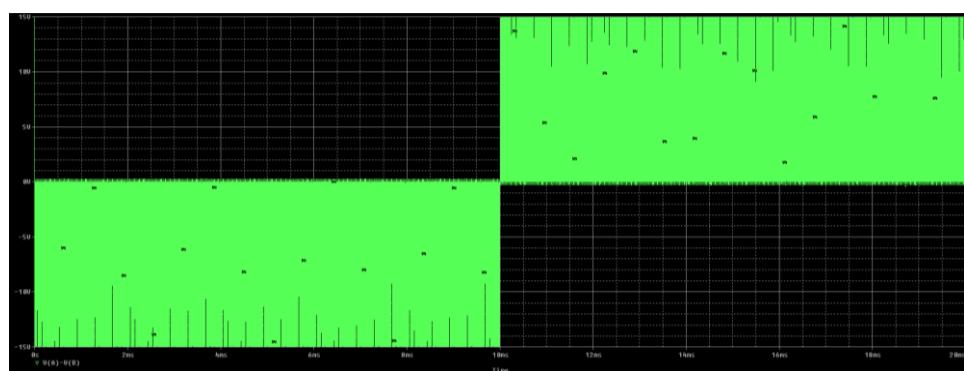


Figura 5.17: Tensión a la salida del puente inversor.

Como podemos ver en la figura 5.17 la tensión en la carga oscila entre mas y menos la tensión de alimentación del puente con una frecuencia igual a la frecuencia de la señal PWM. Sin embargo, hemos de tener en cuenta que la PWM generada en PSPICE no varía el ciclo de trabajo, es decir, se trata de una señal cuadrada de 50 kHz. No obstante, si analizamos el espectro de la señal vemos que aparecen varios armónicos. (Ver Figura siguiente)

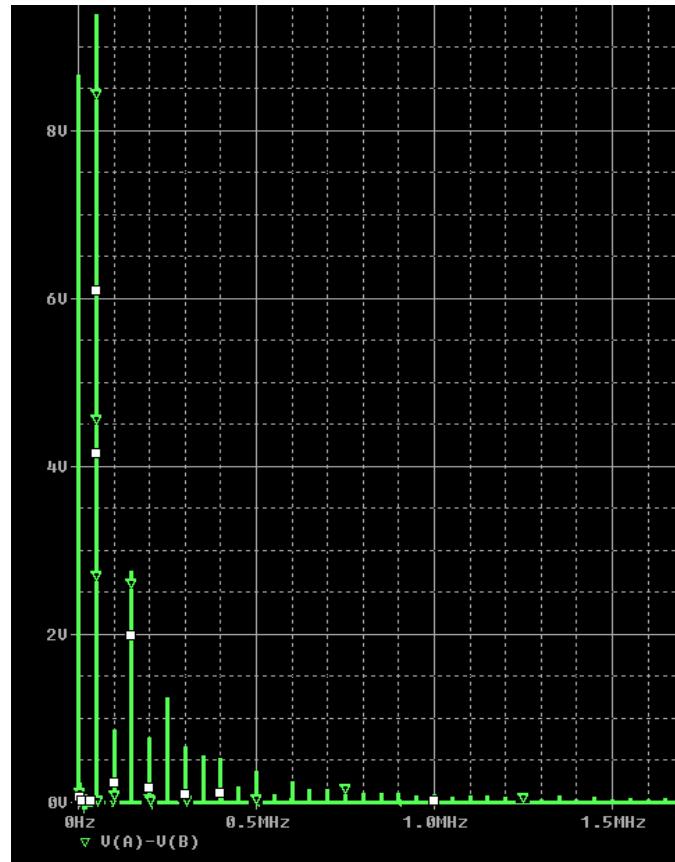


Figura 5.18: Espectro de la tensión obtenida con una carga resistiva.

Si observamos las componentes en frecuencia de la Figura anterior vemos que aparecen bastantes armónicos que no nos interesan. Por ello deberemos realizar un filtrado que nos permita obtener la componente en frecuencia que estamos buscando, que es la componente de 50 Hz, es decir, la componente con una frecuencia igual a la frecuencia de la red eléctrica. Podríamos realizar un filtrado RC como vimos en la figura 5.10, pero con este tipo de filtrado perdemos potencia, por lo que realizaremos un filtrado con un circuito resonante LC.

## 5.8 Filtro LC

A continuación debemos implementar un filtro para conseguir quitar todas las componentes de alta frecuencia y quedarnos solo con la componente de 50 Hz. Podríamos utilizar un filtro RC, pero este tipo de filtro consumen potencia y a nosotros no nos interesa perder potencia, por ello vamos a utilizar un filtro LC [33].

Un circuito LC es un circuito formado por una bobina L y un condensador C en el que para una determinada frecuencia se produce un fenómeno de resonancia eléctrica, a la cual se llama frecuencia de resonancia, de manera que la amplitud de esa frecuencia es incrementada mientras que el resto de frecuencias son atenuadas. Esto se debe a que a la frecuencia de resonancia la reactancia inductiva (parte imaginaria de la impedancia de la bobina) es igual a la reactancia capacitativa (parte imaginaria de la impedancia del condensador) ( $X_C = X_L$ ). Por lo tanto, la impedancia será mínima e igual a la resistencia óhmica. En la figura siguiente podemos ver la función de transferencia de un filtro LC:

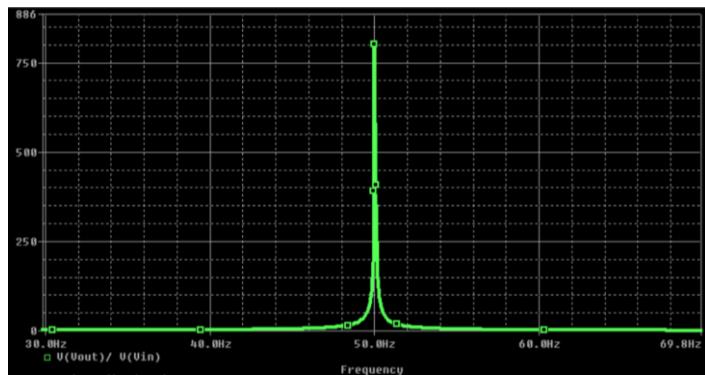


Figura 5.19: Función de transferencia del filtro LC.

La frecuencia de resonancia en un circuito LC viene dada por la siguiente expresión:

$$\omega_0 = \frac{1}{\sqrt{LC}} \quad (\text{eq 5.1})$$

Si queremos que nuestro circuito resuene a 50 Hz, entonces:

$$\text{Si } L = 220 \mu H \rightarrow C = \frac{1}{L * \omega_0^2} = \frac{1}{220 \mu H * (2 * \pi * 50 \text{ Hz})^2} \rightarrow$$

$$C = 46 \text{ mF}$$

Según la teoría necesitarías un condensador de unos 46 mF para filtrar a 50 Hz exactamente, pero tras simular y comprobar varios valores hemos concluido que basta con filtrar con una frecuencia de corte mucho menor que 50 kHz, por lo que podemos obtener un filtrado adecuado con un condensador del siguiente valor:

$$C = 1 \mu F$$

Si pudiéramos conseguir un condensador mayor, por ejemplo, de 1mF o 10 mF también podríamos utilizarlo.

### 5.8.1 Simulación del inversor con filtro LC

A continuación veremos la simulación del inversor completo y comprobaremos si obtenemos la señal que esperamos a la salida que esperamos. En la siguiente figura tenemos el esquemático del inversor con el filtro LC a la salida y la carga.

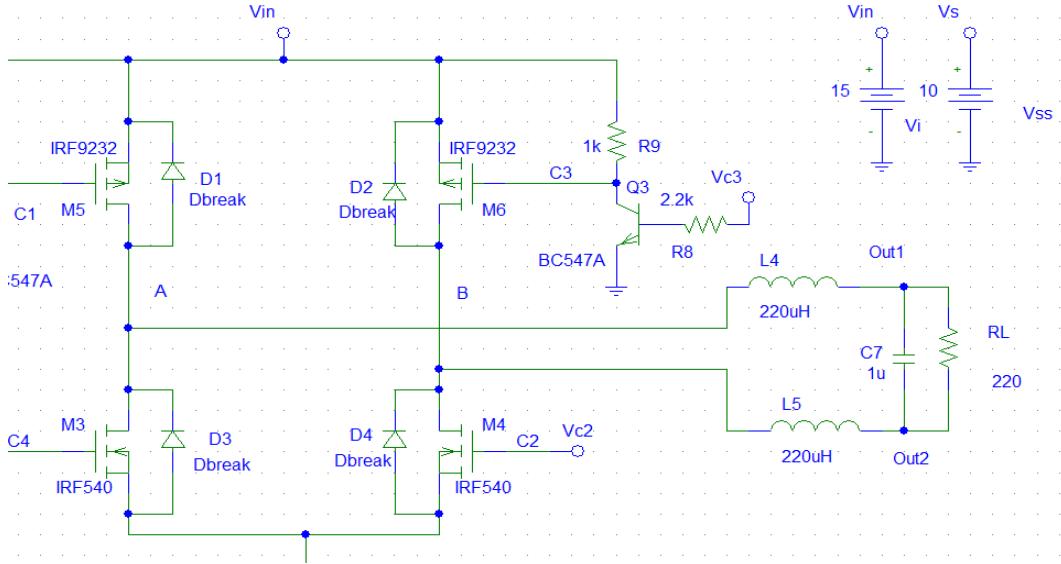


Figura 5.20: Esquemático del inversor con filtro.

Podemos ver los resultados de la simulación en la siguiente figura:

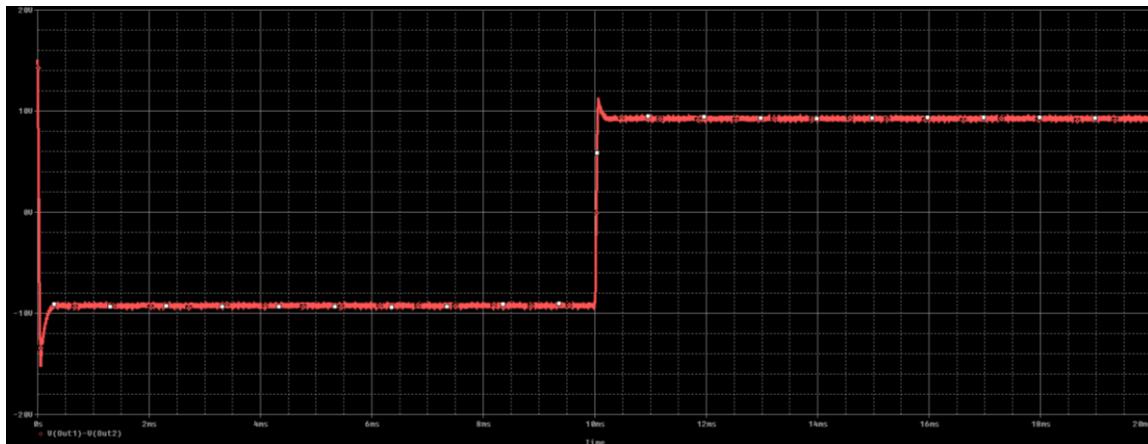


Figura 5.21: Señal obtenida a la salida del filtro.

Lo que vemos en la figura anterior es la señal obtenida a la salida del filtro, la cual tiene una frecuencia de 50 Hz, o un periodo de 20 ms, pero es una señal cuadrada. Como ya comentamos anteriormente la señal PWM que generamos en PSPICE no varía el ciclo de trabajo, es decir es una señal cuadrada de 50 kHz, por eso la señal que obtenemos a la salida del filtro es una señal cuadrada. En la práctica, la señal PWM que generaremos con Arduino si variará el ciclo de trabajo con lo que la señal que obtengamos tras el filtrado sí será senoidal, ya que estará modulada en el ancho de pulso, como vimos en la figura 5.10, en la que filtrábamos la señal PWM con un filtro RC de prueba.

## 5.9 Generación de la señal PWM con Arduino

Como ya vimos en el capítulo anterior, para generar una señal PWM con el Timer 1, era muy sencillo puesto que podíamos definir el valor del TOP del contador, pero este registro solo está disponible para el timer 1, el cual ya está en uso, por ello utilizaremos el timer 2, con el que generaremos la señal de 50 kHz y de 50 Hz mediante el uso de la interrupción por desbordamiento de dicho timer. Para llevar a cabo esto configuraremos el timer 2 para que se desborde cada 20  $\mu$ s (cada periodo de la señal PWM), para lo que necesitaremos configuraremos el timer 2 en el modo fast PWM, cuyo diagrama temporal podeos ver en la figura siguiente.

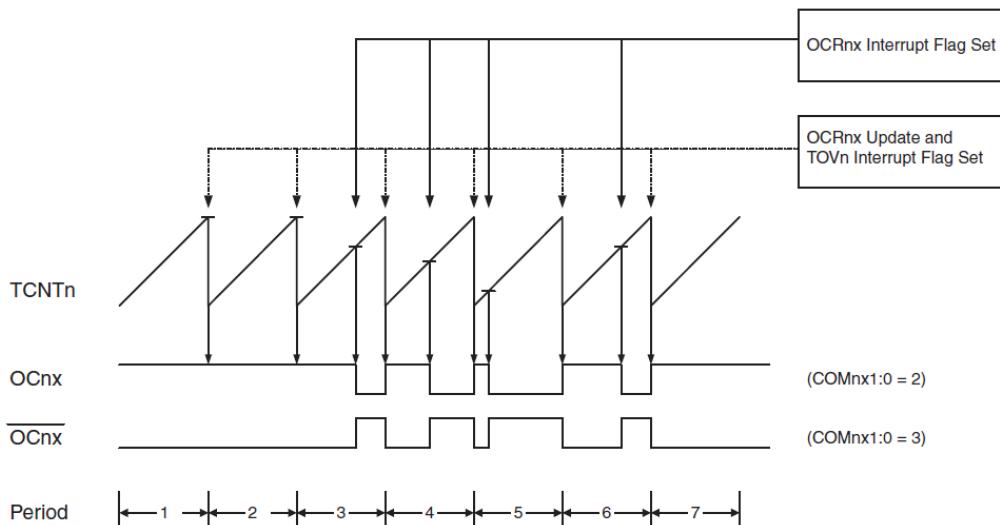


Figura 5.22: Diagrama temporal del modo Fast PWM del timer 2 de Arduino.

En este modo el timer se reinicializa cuando alcanza el valor máximo, para hallar el valor del máximo que nos proporcione una duración del timer de 20  $\mu$ s, hacemos uso de la siguiente expresión:

$$F_{OCnxPWM} = \frac{f_{clk\ i/o}}{N * OCR2A} \quad (eq\ 5.2)$$

Si configuramos el prescaler a  $N = 8$  y, sabiendo que se trata de un contador de 8 bits, por lo que el valor de OCR2A solo puede llegar hasta 258, entonces el valor de OCR2A lo podemos obtener despejando en la expresión anterior de la siguiente manera:

$$OCR2A = \frac{f_{clk\ i/o}}{N * F_{OCnxPWM}} = \frac{16 \cdot 10^6}{8 \cdot 50 \cdot 10^3} = 40$$

Es decir, en el modo FastPWM del timer 2, con prescaler 8 y el registro OC2A = 40 conseguimos que el timer se reinicializa cada 20  $\mu$ s. Una vez hayamos configurado los parámetro tenemos que programar la rutina de interrupción. Para ello, lo que haremos será asignar el valor del ciclo de trabajo más bajo al inicio, después la rutina debe contar la veces que desborda el timer (cada vez que transcurren 20  $\mu$ s). Cuando el contador llegue a 50 habrá transcurrido 1 ms y será el momento de asignar un nuevo valor de ciclo de trabajo, y esto lo iremos haciendo cada 1ms hasta llegar a 10 ms.

Por otro lado, para generar la señal de 50 Hz, tenemos que tener en cuenta que ambas señales deben estar sincronizadas, es por ello que tenemos que generar la señal de 50 Hz con el mismo timer, para eso, lo que haremos será utilizar un segundo contador que contará cada 1ms, y cuando el contador llegue a 10 (hayan pasado 10 ms), entonces variaremos el estado de uno de los pines de Arduino, generando un señal de cuadrada de 50 Hz. En la figura 5.10 ya vimos una imagen en la que aparecen las señales de 50 kHz y 50 Hz que hemos generado con Arduino. A continuación vamos a ver una imagen en la que aparecen las señales de control del puente a la salida de la lógica que diseñamos anteriormente.

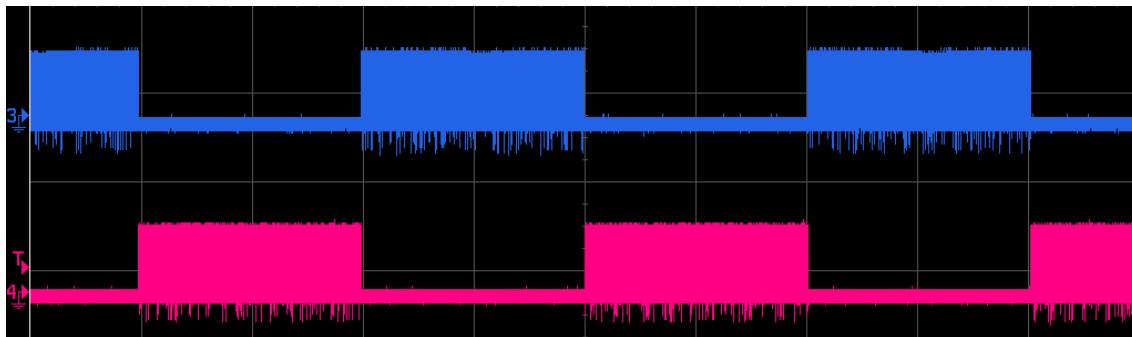


Figura 5.23: Señales de control del puente inversor.

En la figura anterior cada una de las señales conmutará un lado del puente, pero recordemos que la señal que conmuta al transistor P de la parte alta, requiere una etapa previa que niegue el valor de la señal de control. En este punto, si conectamos todo los componentes del inversor y colocamos una carga resistiva, ya conseguiríamos una señal invertida. Pero antes de ver la señal que obtenemos en la carga tenemos que hablar sobre ciertas precauciones que tenemos que tener a la hora de conmutar el puente

## 5.10 Precauciones con cargas inductivas

Como ya comentamos anteriormente, si conectamos una carga inductiva, como un transformador, hay que respetar un tiempo de off de las señales de control entre cada estado de conmutación. Este tiempo de off permitir que el núcleo de la bobina del primario de desmagnetice en cada conmutación, es decir, cuando uno de los lados del puente deja de conducir, debe haber un tiempo en el que el puente este desactivado antes de que el otro lado comience a conducir. Durante este tiempo el núcleo del primario, que está magnetizado, se desmagnetiza a través de los diodos en antiparalelo que hemos colocado en los MOSFET, de manera que la energía almacenada en el transformador fluye en forma de corriente a través de estos diodos y vuelve a la fuente.

Para conseguir este tiempo en off, solo tenemos que ajustar la señal PWM para que esté a cero durante 1ms a cero cada 9ms. Pero en realidad esto no es tan trivial, ya que las señales están mezcladas con puertas NAND y esto puede hacer que la señal de salida sufra retardos que deterioren las señales. Para comprender mejor el algoritmo véase el código de Arduino adjunto en la documentación del proyecto. A continuación vamos a ver cómo quedan las señales con el tiempo de OFF.

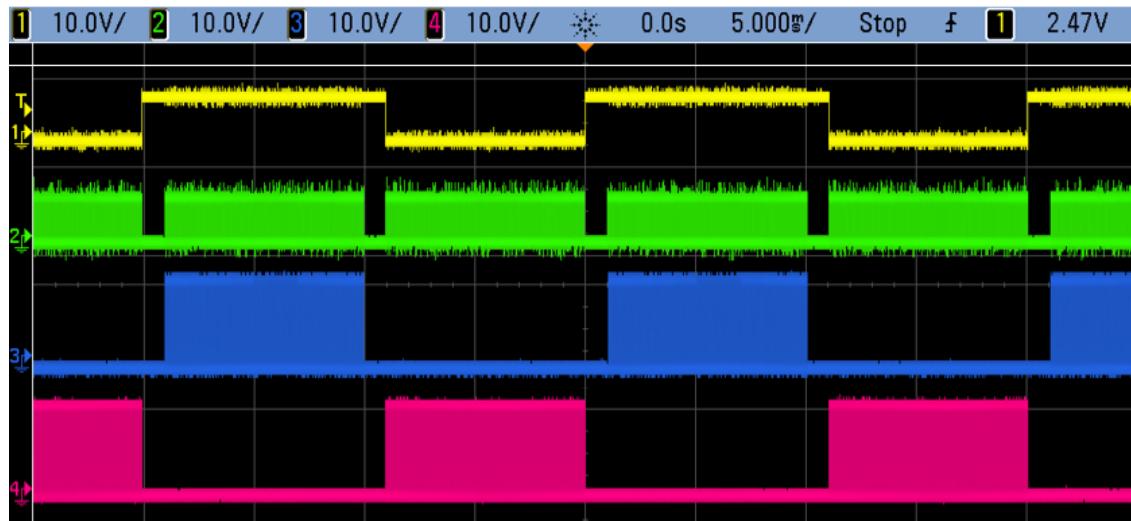


Figura 5.24: Señales de control con tiempo de OFF.

En la figura anterior podemos ver la señal de 50 Hz, la señal PWM con desconexión de 1 ms cada 9 ms, y las señales de control de cada uno de los lados del puente que. En este punto ya tenemos nuestras señales de control listas para que funcione nuestro dispositivo. Lo siguiente que vamos a ver es la señal que obtenemos en la carga. Véase la figura 5.25.

### 5.11 Señales de salida obtenidas con una carga resistiva

A continuación vemos una captura tomada con el osciloscopio donde se representado la formada de onda en la carga del inversor:



Figura 5.25: Señal obtenida en la carga.

En la figura 5.25 lo que vemos son las señales en cada uno de los terminales de la resistencia de carga (en amarillo y verde), la diferencia entre ambos canales y la señal de 50 Hz que nos sirve para el disparador del osciloscopio. Como vemos, la señal obtenida en la carga es justamente la que esperábamos. La tensión oscila entre más y menos el valor de la tensión de la entrada del inversor, excepto en los tiempos de off, en los que la salida se anula. Tenemos que mencionar que en la salida diferencial no vemos la señal PWM porque el canal matemático del osciloscopio no tiene demasiada resolución, pero si hacemos zoom si conseguiremos ver la señal PWM en la carga.

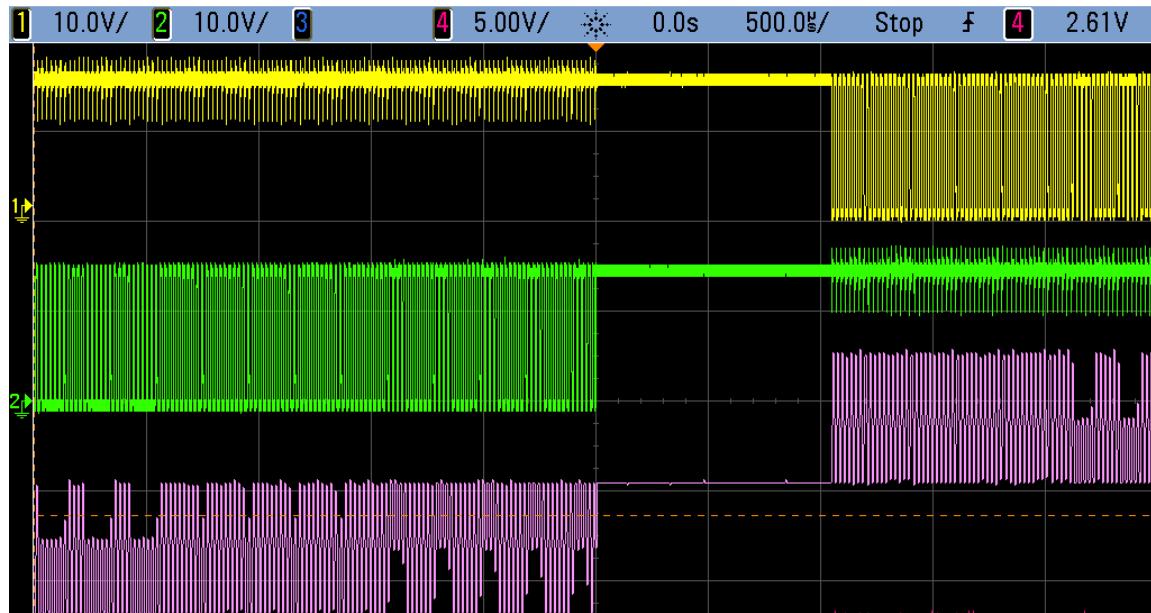


Figura 5.26: Zoom de la señal obtenida en la carga.

Como vemos en la figura anterior, la señal de salida en la carga contiene la PWM excepto en los tiempos de off. Lo que haremos a continuación será mostrar cómo queda la señal de salida después de filtrarla, para lo que utilizaremos el filtro LC del que hablamos anteriormente.

## 5.12 Señal a la salida del filtro

Si finalmente conectamos el filtro LC a la salida del inversor obtenemos la siguiente señal de salida. Recordamos que al final del capítulo podemos encontrar una imagen de la placa donde están remarcados los componentes que conforman esta etapa. Recordamos también que la señal que obtenemos no podría ser inyectada en la red eléctrica ya que para poder hacer esto, la frecuencia instantánea de nuestra señal debería ser exactamente igual a la frecuencia de la red, es decir, la frecuencia de la red debería ser leída en tiempo real. No obstante, la señal de 50 Hz que obtenemos aquí sí podría ser utilizada en dispositivos que funcionen con corriente alterna, como un PC, una lámpara o el cargador de un teléfono móvil. Mencionado esto, vamos a ver la señal obtenida a la salida del filtro. También recordar que el condensador que estamos utilizando en el filtro no es el más adecuado, por lo que nos aparecerán ciertas componentes que podremos eliminar por otros medios.

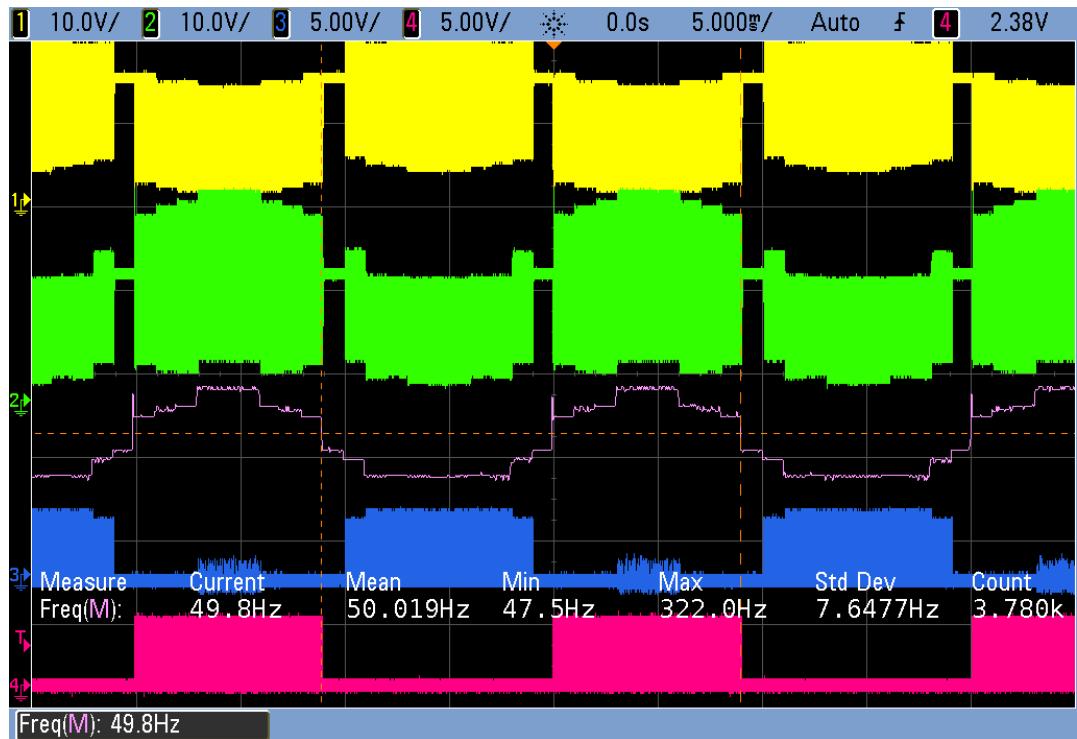


Figura 5.27: Señal obtenida a la salida del filtro.

En la figura anterior podemos ver la señal a salida en el filtro. Esta señal ya no contiene la PWM, es decir, se trata de una señal de 50 Hz oscilante, con aspecto casi sinusoidal, debido a que la señal aún conserva algunas componentes que no han sido filtradas, pero podemos conseguir mejor la señal si variamos el valor de los ciclos de trabajo, para que no aparezcan esos escalones que degradan la señal, como vemos en la figura siguiente.

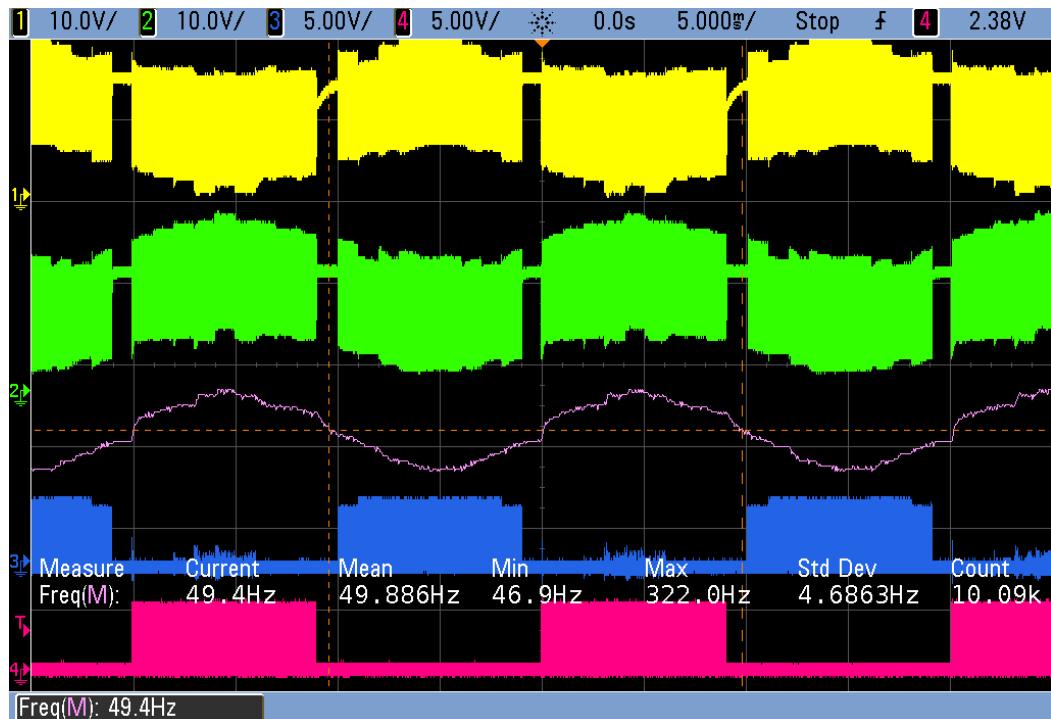


Figura 5.28: Señal de salida del filtro LC mejorada

### 5.13 Distribución de los componentes

Como en el capítulo anterior, el último apartado de este capítulo lo dedicamos a la visualización de los componentes que forman esta etapa.

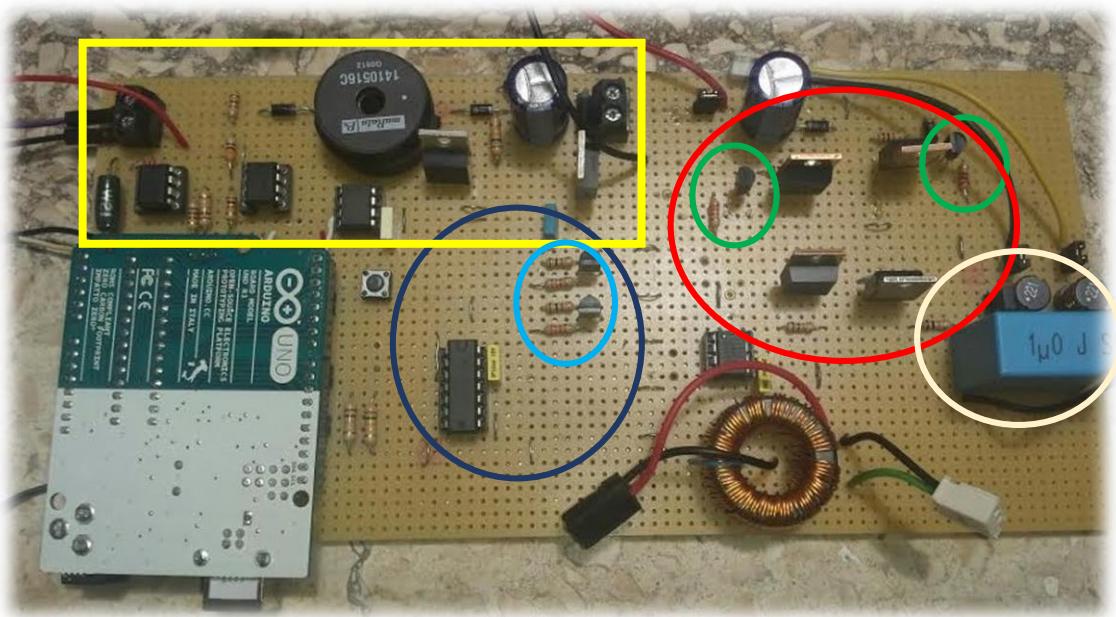


Figura 5.29 Situación de los componentes del inversor.

Rodeado de color rojo podemos ver el inversor de puente completo formado por cuatro transistores MOSFET, en el que los dos transistores de arriba de canal P y los de abajo son de canal N, en color verde vemos las etapas que niegan la señal de control para los transistores de canal P, con un círculo azul oscuro vemos los componentes que conforman la lógica digital que genera las señales de control de cada interruptor; CD74H00 del que hablamos en el apartado anterior y los amplificadores bipolares que aumenta la corriente de salida del integrado en azul claro, y por último, en color blanco podemos ver el filtro LC, con sus dos bobina de  $220\ \mu\text{H}$  y el condensador de  $1\ \mu\text{F}$ . Como ya sabemos, el recuadro amarillo marca el convertidor DC-DC elevador.

En este punto ya hemos visto, por un lado cómo diseñar los componentes hardware de nuestro proyecto, el convertidor DC-DC elevador y el convertidor DC-AC (inversor), y por otro lado hemos visto cómo implementar los algoritmos que generan las señales de control que nos permiten manipular estos componentes. Si hay algo que no ha quedado claro del software de control puede verse el código fuente adjunto en la documentación del proyecto. Lo que vamos a hacer a continuación es explicar cómo explotar los datos que conseguimos de las etapas previas. Para ello necesitaremos algún tipo de comunicación entre PC y Arduino que nos permita recibir los datos, representarlos y ajustar el sistema de manera remota.

---

## **6. Comunicación Arduino-PC**

En este capítulo vamos a hablar sobre cómo se realiza la comunicación entre Arduino y PC, desde los módulos hardware hasta las aplicaciones implementadas para que se establezca dicha comunicación, así como los protocolos y las normas que intervienen en ella. Todo esto permitirá a nuestro dispositivo enviar información a distancia sobre el estado general del sistema cuando sea solicitada. Además vamos a ver cómo crear una interfaz de usuario con MATLAB que permita al usuario comunicarse con el PC y controlar el sistema de una manera sencilla, mediante el uso del ratón del PC para hacer que la interacción humano-máquina sea lo más cómoda y eficiente posible.

---

## 6.1 Comunicación serie

La comunicación serie [34] es el proceso de envío de datos bit a bit de forma secuencial y a una misma velocidad de transmisión, que se mide en bps o baudios, por un canal de comunicaciones. A diferencia de la comunicación en paralelo, en la que los bits de cada símbolo se envían al mismo tiempo. La ventaja de la comunicación serie es que necesita un número reducido de líneas de transmisión mientras que la comunicación en paralelo requiere al menos tantas líneas como bits tenga la información a transmitir. Además, la comunicación en paralelo presenta problemas de interferencia o desincronización. No obstante, para una misma frecuencia de transmisión, la comunicación paralela tiene un mayor rendimiento. A continuación podemos ver un ejemplo de comunicación serie.

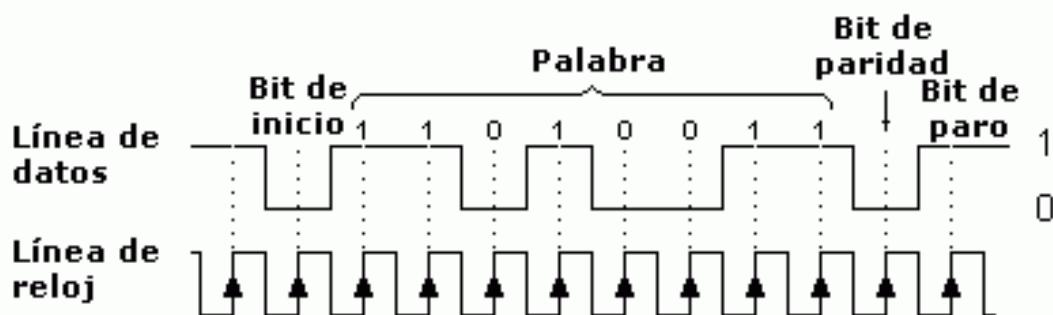


Figura 6.1: Ejemplo de comunicación serie. [35]

La comunicación serie puede ser de varios tipos:

- Simplex: Emisor y receptor están perfectamente definidos y la comunicación es unidireccional.
- Half duplex: Un canal es compartido por emisor y receptor, pero no al mismo tiempo.
- Full dúplex: Emisor y Receptor sí pueden emitir y recibir simultáneamente.

En el ejemplo de la figura anterior podemos ver una comunicación simplex en la que se usan dos líneas de transmisión; una para la señal de reloj, necesaria para enviar los bits secuencialmente cada flanco de subida, y una línea de datos, cuyo protocolo consiste en mandar un bit de inicio “0”, una palabra de 8 bits que contiene la información que queremos transmitir codificada con el estándar ASCII, un bit de paridad para comprobar si ha habido errores en la transmisión y, por último, un bit de parada “1”.

Como podemos ver la comunicación serie necesita una señal de reloj para la transmisión de información, por esta razón, conforme los procesadores han ido incrementando su velocidad, la comunicación serie ha ido desplazando a la comunicación paralelo en un gran número de aplicaciones.

## 6.2 Puerto Serie

El puerto serie es un interfaz físico o virtual que necesita al menos dos conectores para establecer una comunicación serie de datos. Un ordenador convencional puede disponer de varios tipos de puertos serie, los más conocidos son el USB (Universal Serial Port) y el ya casi olvidado (RS-232).

### 6.2.1 USB (Universal Serial Bus)

Es un bus estándar industrial que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadores, periféricos y dispositivos electrónicos. A continuación podemos ver el patillaje del USB tipo A y tipo B, que son los conectores que utilizan PC y Arduino UNO respectivamente

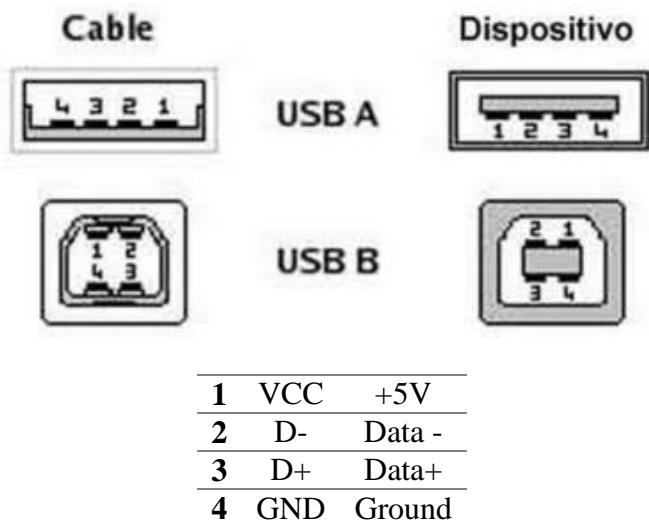


Figura 6.2: Patillaje USB A y USB B. [36]

Como podemos ver, el puerto USB tiene dos canales de datos para enviar y recibir información y dos canales de alimentación, que nos permiten alimentar el microcontrolador.

### 6.3 USRAT (Universal Serial Asynchronous Receiver-Transmitter)

USART es un dispositivo de comunicación serial full dúplex que el microcontrolador ATMEGA 328P de Arduino UNO tiene integrado dentro del chip, eso permite al microcontrolador realizar otras tareas mientras está recibiendo datos que puede ir guardando en un buffer de 64 kbytes. La conexión USB con el ordenador la realiza el chip externo FTI-USB que recibe datos desde el microcontrolador en formato TTL, en el que un “1” lógico se representa con una señal de 5V y un “0” lógico se representa con el nivel 0 de tensión (Podemos ver un enlace TTL en la figura siguiente), y el chip envía estos datos por USB a través de un driver de software que ve a Arduino como un puerto serie.



Figura 6.3: Enlace TTL, no recomendable para distancias mayores de 5m. [37]

### 6.3.1 Comunicación USART

USART tiene un generador de reloj para el transmisor y otro para el receptor con diferentes modos de operación y un generador de velocidad de transmisión, ambos son necesarios para una comunicación serie. El formato de la trama de datos que envía Arduino es la siguiente:

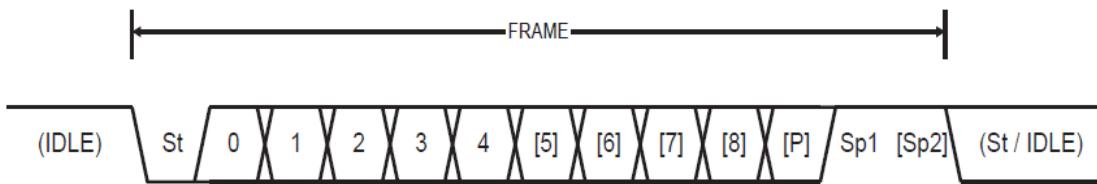


Figura 6.4: Formato de la trama. [21]

- St: Start bit, siempre “0”.
- (n): Bits de datos (0 hasta 8).
- P: Bit de paridad.
- Sp: Stop bit, siempre “1”. Opcionalmente puede tener dos bits de parada.
- IDLE: No se transmite en la línea Rx o Tx. Debe ser siempre “1”.

Ya que conocemos todos estos aspectos relacionados con la comunicación serie, lo siguiente es implementar un software que nos permite enviar datos desde Arduino al PC y viceversa. De esta manera podremos representar los datos que obtengamos con Arduino y tener una visión más clara de los que está ocurriendo en nuestro sistema fotovoltaico. Pero antes de eso vamos a hablar sobre una característica del controlador que nos será muy útil a la hora de comunicarnos con él.

### 6.3.2 Interrupción por USART

Las interrupciones nos permiten que el microcontrolador realice una tarea concreta cuando ocurre un determinado evento, en este caso, cuando recibe algo por el puerto serie. Para programar esta interrupción podemos hacerlo de dos formas:

- Configurar el vector de interrupción adecuado: ISR(USART\_RX\_vect) {}
- Utilizar la función serialEvent()

En nuestro caso nos hemos decantado por la segunda opción. La función serialEvent() se ejecuta cada vez que se recibe un dato por el puerto serie. Cuando se produce la interrupción se inicia una rutina que hace uso de las funciones serial.available() y serial.read() para leer el dato. (Ver código fuente adjunto en la documentación). Cuando el dato está disponible se hace un control para comprobar si es el carácter que hemos definido para realizar la calibración, un “1”, en tal caso, comienza a calibrar el sistema y a enviar las muestras al PC para poder representarlas gráficamente, en caso contrario el sistema no realiza ninguna acción. En este punto MATLAB ya estará esperando para recibir muestras. Esto lo explicaremos en breve.

## 6.4 Librería *serial* de Arduino

La librería *Serial* permite la comunicación serial con el PC u otros dispositivos. Las principales funciones incluidas en esta librería son:

- `Serial.begin(rate)`: Permite establecer la velocidad en bits por segundo (baudios).
- `Serial.end()`: Desactiva la comunicación serie.
- `Serial.available()`: Devuelve el número de bytes disponibles para ser leídos.
- `Serial.read()`: Lee los datos entrantes por el puerto serie.
- `Serial.println(data,format)`: Envía datos en un formato determinado.

Utilizando estas funciones vamos a desarrollar una aplicación que haga que Arduino comience a tomar medidas cuando le enviamos un determinado dato por el puerto serie y que las vaya enviando al PC.

## 6.5 Comunicación serie con MATLAB

MATLAB es un lenguaje de programación de alto nivel y un entorno interactivo que nos permite realizar tareas de cálculo numérico, análisis y visualización de datos, programación y desarrollo de algoritmos y aplicaciones. Para nuestro proyecto implementaremos una aplicación que nos permita representar las curvas I-V y P-V de un panel fotovoltaico y calcular el punto de máxima potencia y el factor de llenado de dicho panel. Para ello recibiremos muestras a través del puerto serie. Las funciones que utiliza MATLAB para la comunicación serie son las siguientes:

- `instrfind()`: Devuelve un puerto serie valido.
- `obj = serial('port')`: Crea un objeto “puerto serie” asociado a un puerto.
- `puerto_serial.BaudRate = Rate`: permite definir la velocidad de transmisión.
- `fopen(obj)`: Abre el puerto serie asociado a un objeto.
- `fprintf(obj,'format',dato)`: Envía un dato con un formato determinado.
- `fscanf(obj,'format')`: Permite leer un dato por el puerto serie
- `fclose(obj)`: Cierra el puerto serie.

Cuando se ejecuta la instrucción `fscanf()`, el PC espera hasta que recibe el dato por el puerto serie, y esta función siempre se ejecuta antes de que Arduino transmita, ya que el PC tiene una frecuencia de reloj mucho más rápida y cada instrucción en MATLAB se ejecuta más rápido que las de Arduino. Para comprender mejor el funcionamiento del software es mejor leer el código fuente detenidamente, ya que este tiene comentarios aclaratorios en cada instrucción. En líneas generales, lo que hace nuestra aplicación es crear un gráfico y enviar un dato de control, un 1, a Arduino para que empiece a tomar muestras de la tensión y la corriente y las vaya transmitiendo al PC. Cuando Arduino recibe el dato de control comienza a variar el ciclo de trabajo de la señal PWL que controla el convertidor DC-DC elevador, desde 0 hasta el valor máximo, que como ya explicamos es 33 (un incremento del 3% en cada salto) y en cada valor del ciclo de trabajo se comprueba si la potencia es máxima. A continuación tenemos un diagrama de flujo que nos puede dar una idea de las tareas que realiza la aplicación de control.

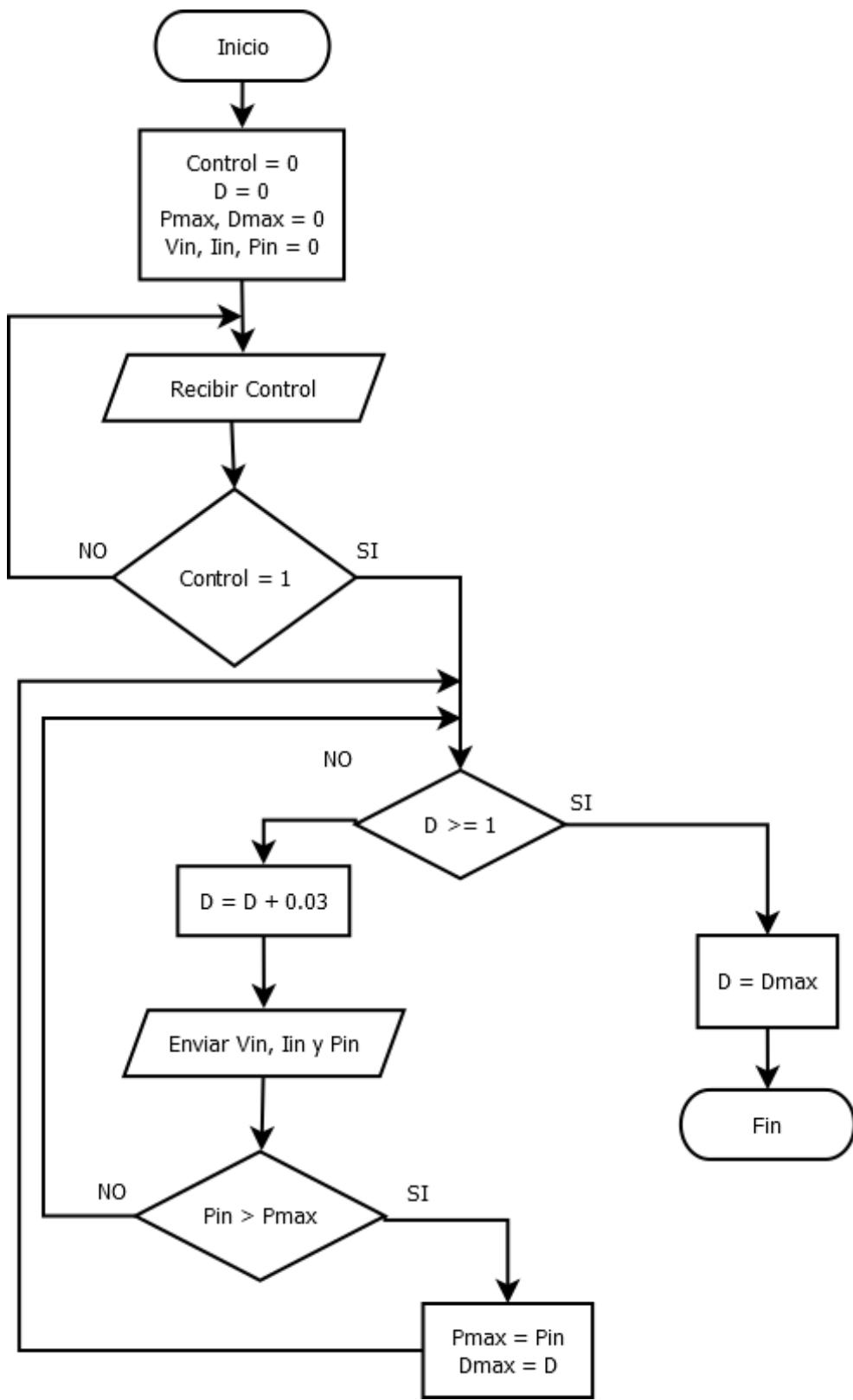


Figura 6.5: Diagrama de flujo de la aplicación de control.

En el siguiente apartado vamos a ver cómo crear una interfaz de usuario con MATLAB que facilite la interacción con el dispositivo y nos permita visualizar los resultados.

## **6.6 GUI de MATLAB**

La GUI [38] (también conocidas como interfaces gráficas de usuarios o interfaces de usuario) permiten un control sencillo de las aplicaciones de software, lo cual elimina la necesidad de aprender un lenguaje y escribir comandos a fin de ejecutar una aplicación.

Las apps de MATLAB son programas autónomos de MATLAB con un frontal gráfico de usuario GUI que automatizan una tarea o un cálculo. Por lo general, la GUI incluye controles tales como menús, barras de herramientas, botones y controles deslizantes. Muchos productos de MATLAB, como Curve Fitting Toolbox, Signal Processing Toolbox y Control System Toolbox, incluyen apps con interfaces de usuario personalizadas. También es posible crear apps personalizadas propias, incluidas las interfaces de usuario correspondientes, para que otras personas las utilicen.

## **6.7 Diferentes caminos para crear de una GUI de MATLAB**

En este apartado vamos a ver que para crear interfaz de usuario con MATLAB existen dos maneras:

- Creación de una GUI de forma interactiva

GUIDE (entorno de desarrollo de GUI) proporciona herramientas para diseñar interfaces de usuario para Apps personalizadas. Mediante el editor de diseño de GUIDE, es posible diseñar gráficamente la interfaz de usuario. GUIDE genera entonces de manera automática el código de MATLAB para construir la interfaz, el cual se puede modificar para programar el comportamiento de la app.

- Creación de una GUI de forma programática

A fin de ejercer un mayor control sobre el diseño y el desarrollo, también se puede crear código de MATLAB que defina las propiedades y los comportamientos de todos los componentes. MATLAB contiene funcionalidad integrada que le ayudará a crear la GUI para su app de forma programática. Cabe la posibilidad de agregar cuadros de diálogo, controles de interfaz de usuario y contenedores.

## **6.8 Desarrollo de una interfaz de usuario**

A continuación vamos describir los pasos a seguir para crear una interfaz de usuario interactiva que nos permita representar datos con un solo clic independientemente del PC al que se conecte nuestro dispositivo.

### **6.8.1 Entorno de desarrollo**

Para iniciar el entorno de desarrollo tecleamos el siguiente comando:

```
>>guide
```

Y nos aparecerá la siguiente ventana:

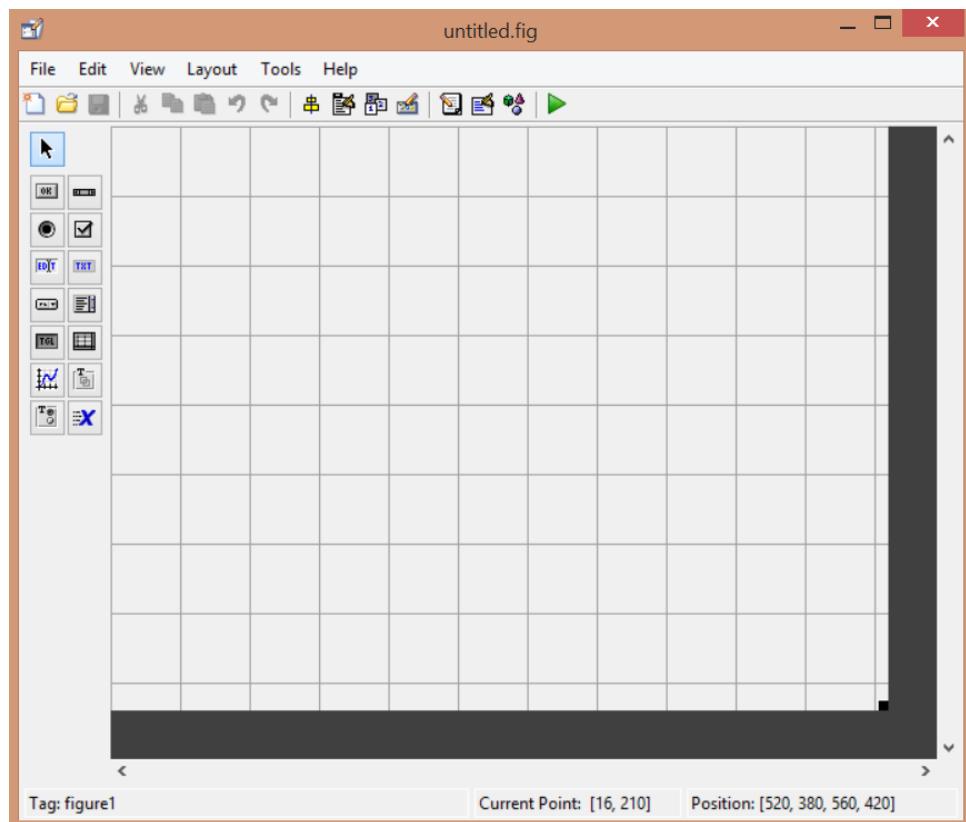


Figura 6.6: Ventana de desarrollo de GUIDE.

En esta ventana podemos añadir diferentes componentes, como botones, ejes, pop-up menús, textos, casillas, sliders, etc. A continuación vamos a mostrar cómo queda esta ventana después de añadir los componentes que necesitamos para nuestro proyecto, y explicaremos cómo funciona cada uno de ellos.

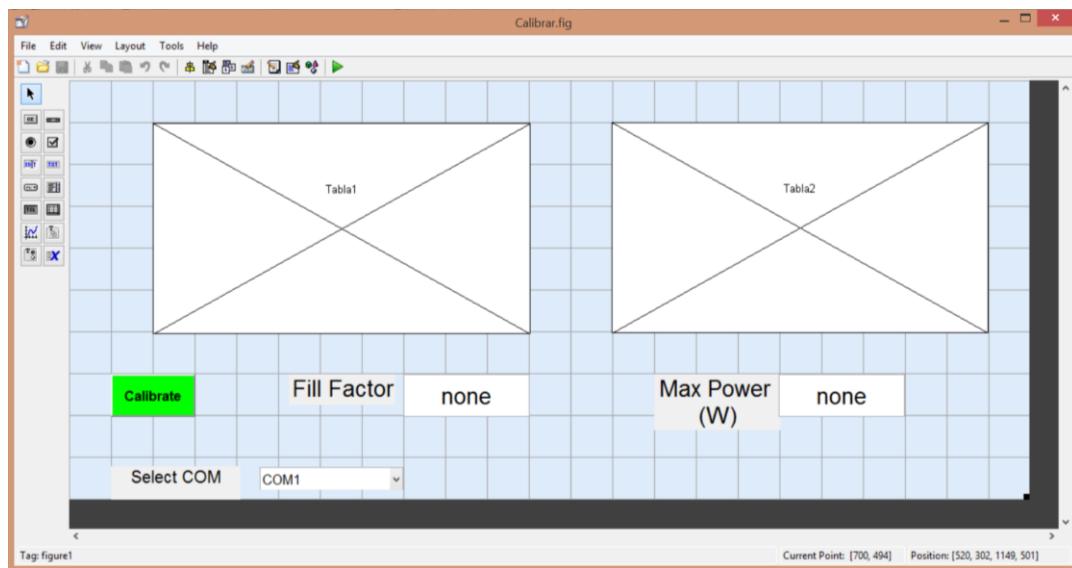


Figura 6.7: Ventana con los componentes necesarios para el proyecto.

Cuando hablamos de la comunicación serie en el capítulo 6, ya comentamos algunas funciones de MATLAB que nos permitían representar datos de manera dinámica. Lo que vamos hacer a continuación es aprender como modificar estas funciones para que los resultado aparezcan en la interfaz de usuario, pero antes de eso tenemos que saber que cuando añadimos componentes en GUIDE se nos generar automáticamente un código fuente en el que aparece la función que realiza cada componente y que podemos modificar a conveniencia.

### 6.8.2 Aplicación a desarrollar

Las funciones que desempeñará la interfaz serán: Seleccionar el puerto COM en el que esta Arduino. Esto es útil porque el puerto en el que se instala Arduino cambia de un PC a otro, y nosotros queremos que nuestra aplicación funcione en cualquier PC. Una vez que seleccionemos el puerto COM pulsaremos el botón calibrar cuya tarea será realizar todas las acciones de control que comentamos anteriormente.

A continuación explicaremos cuáles son las funciones de MATLAB que nos permiten obtener información desde los componentes interactivos y cuál nos permiten acceder a ellos para enviarles información.

Para acceder a la información utilizamos la función get():

```
get(handles.PopupMenu, 'value');
```

handles.PopupMenu hace referencia al componente pop-up menú, en el que cada una de las opciones tiene un valor entero 1, 2, 3... Para poder leer el valor que obtengamos desde este componente tendremos que convertirlo a string para que MATLAB abra el puerto serie adecuado:

```
NCOMstr = num2str(NCOM);  
COM = ['COM', NCOMstr];
```

Para enviar información en un gráfico utilizamos al función axes():

```
axes(handles.Tabla1);
```

handles.Tabla1 se refiere a la primera gráfica. Los plot que hagamos a continuación de esta sentencia se realizarán sobre esta tabla. Si queremos escribir en la otra tabla solo tenemos que llamar a la función axes() haciendo referencia a su identificador.

Por último para escribir los resultados en las casillas de texto utilizamos la función set();

```
set(handles.Mpower, 'String', maxPstr);
```

Ya sabemos cuál sería el identificador de la casilla de texto. Para representar el dato en la casilla de texto dato de la potencia máxima a tipo string:

```
maxPstr = num2str(maxP);
```

Para más información consúltese el código fuente adjunto en la documentación.

## 6.9 Generación de un archivo ejecutable .EXE con deploytool

Hay dos casos en los que compilar tu código puede ser ventajoso:

- Si queremos distribuir el código a personas que no tengan MATLAB instalado
- Trabajar con una versión stand alone no requiere una licencia de MATLAB.

Para compilar nuestra aplicación ejecutamos deploytool

```
>>deploytool
```

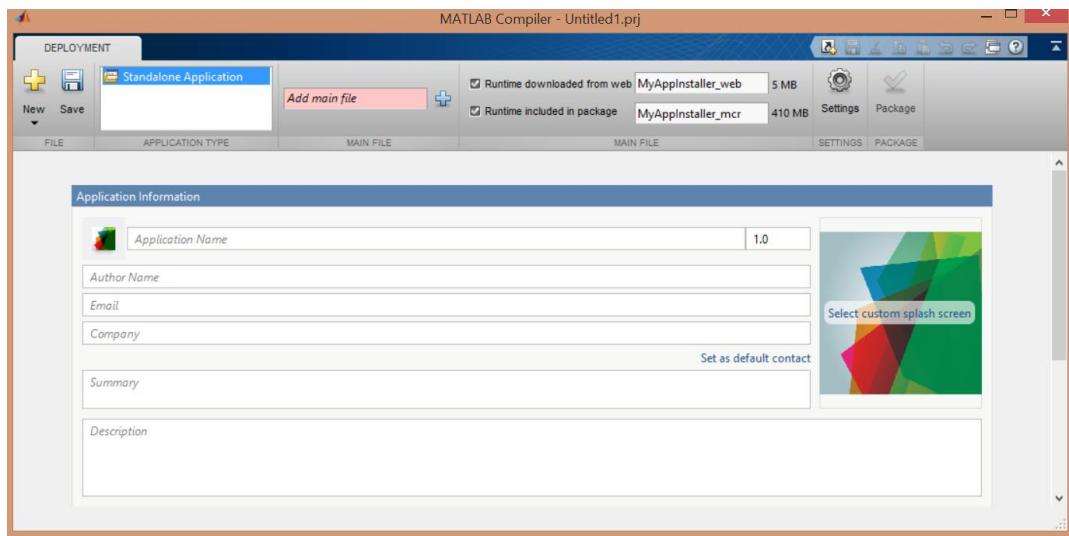
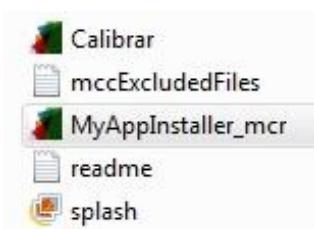


Figura 6.8: Compilador de MATLAB.

El compilador nos permite configurar una serie de parámetros. En primer lugar añadimos el “.m” principal y después añadimos los archivos secundarios, figuras e imágenes. Podemos poner el nombre que queramos a la aplicación, una descripción y un ícono. Un dato importante es que los archivos de runtime necesarios para la ejecución de la aplicación pueden estar incluidos en el package, pueden descargarse desde internet o ambos. Seleccionamos el método de runtime que queramos, en nuestro caso estarán incluidos y hacemos clic en package. El sistema nos generará los siguientes archivos:



Por un lado tenemos el ejecutable, que solo podemos usarlo si tenemos MATLAB instalado, y por otro lado está el instalador. Si ejecutamos el instalador nos aparecerá la ventana de instalación de nuestra aplicación, con la descripción que le hayamos dado:

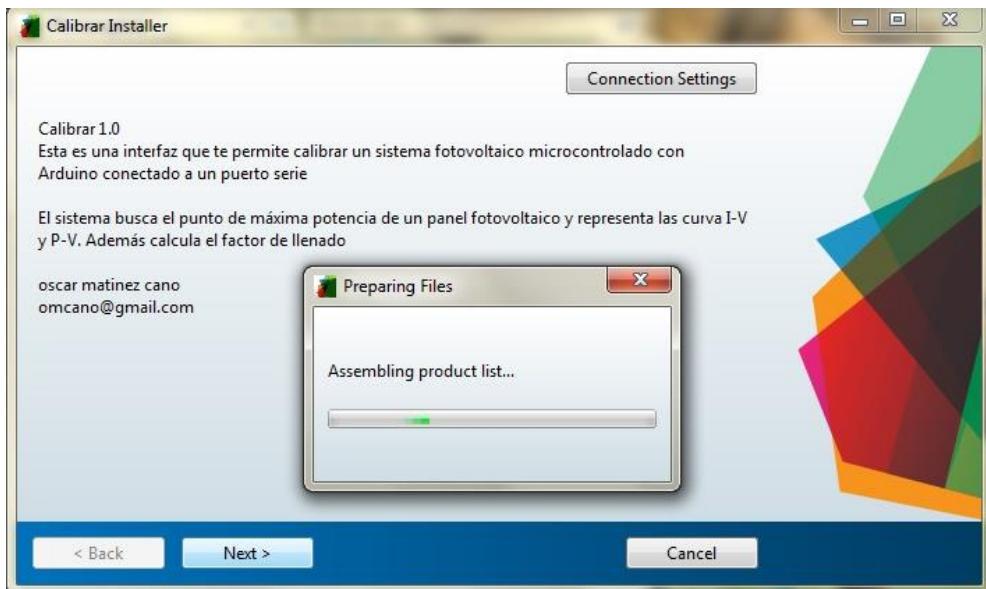


Figura 6.9: Instalación de la aplicación.

Con esto damos por concluido la explicación de los aspectos técnicos de este TFG, en el que hemos aprendido sobre temas relacionados con la energía solar fotovoltaica, electrónica de potencia, sistemas microcontrolados, comunicaciones e interfaces de usuario. Espero que haya sido de utilidad al lector y que haya aprendido tanto como yo. A continuación veremos algunos resultados y los componentes que podríamos añadir a este trabajo en futuros proyectos con el fin de darle una mayor utilidad y aprender más sobre la generación y el control de energía.

---

## **7. Resultados**

En este último capítulo vamos a ver los resultados obtenidos al calibrar el sistema en diferentes momentos del día. Y veremos cómo se visualizan mejor los datos gracias a la interfaz de usuario.

---

A continuación vamos a ver varias gráficas obtenidas tras calibrar el sistema en diferentes momentos del día, la gráfica superior corresponde con la curva I-V y la inferior con P-V.

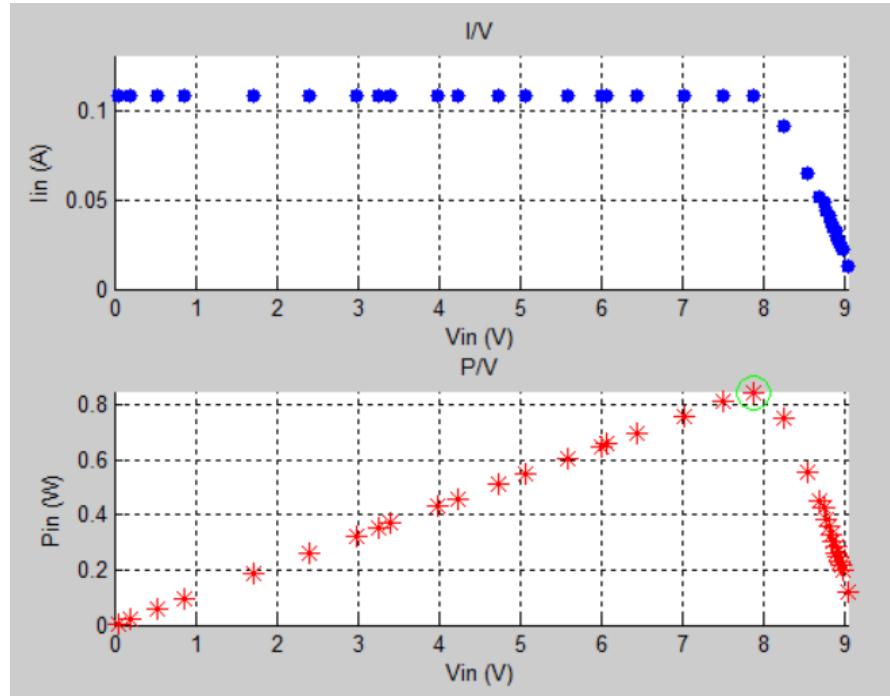


Figura 7.1: Curvas I-V y P-V a las 12:00.

$$\text{FF} = 0.86.$$

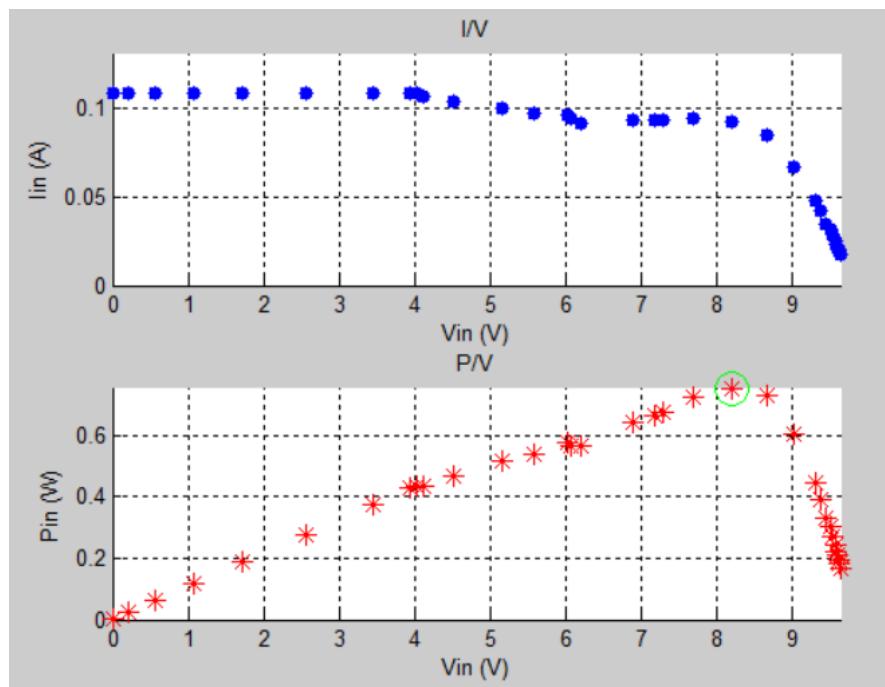


Figura 7.2: Curvas I-V y P-V a las 16:00.

$$\text{FF} = 0.72.$$

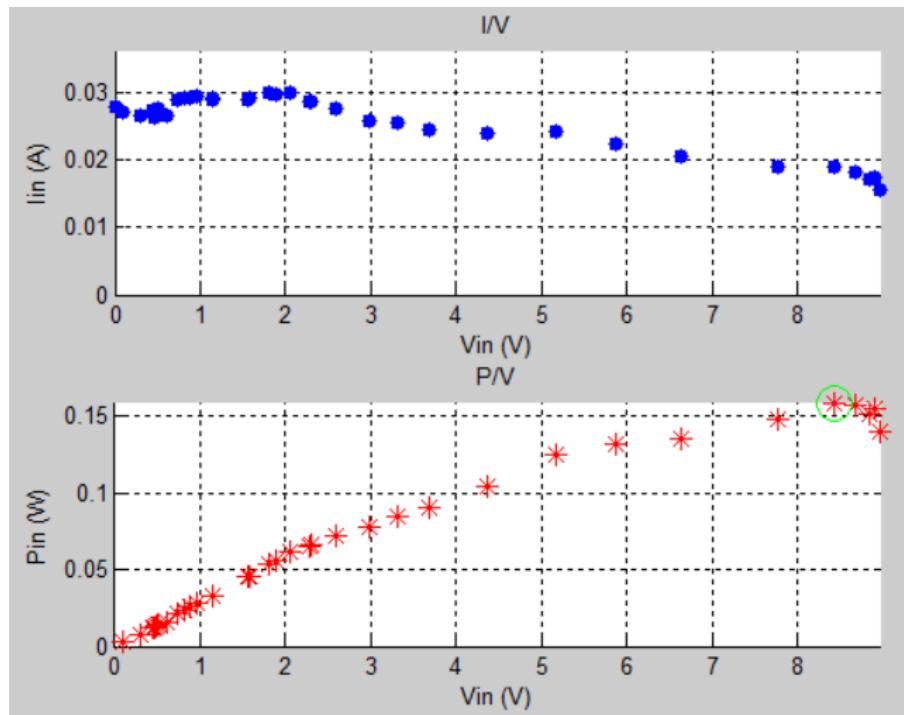


Figura 7.3: Curvas I-V y P-V a las 18:30.

$$\text{FF} = 0.64$$

Como podemos ver en las gráficas, nos aparece remarcado en verde el punto de máxima potencia. Este punto es calculado por MATLAB con los datos que ha recibido, por otro lado Arduino también lo calcula y ajusta el ciclo de trabajo del convertidor que hace que el panel opera a máxima potencia. Podríamos enviar el punto de máxima potencia desde el PC al Arduino y esto ahorraría procesamiento al microcontrolador, pero en principio nuestro microcontrolador funciona correctamente. Si en futuro añadiéramos más carga de procesamiento a Arduino podríamos hacerlo de esa otra manera o incluso calcularlo por separado y comprobar que los resultados concuerden si lo que queremos es mayor seguridad. Nosotros ya comprobamos que el ciclo de trabajo que maximiza la potencia calculado por Arduino y por MATLAB coinciden.

Hasta aquí ya tenemos todo lo necesario para que nuestra aplicación funcione correctamente y nos permita ajustar diferentes paneles en diferentes instantes del día, de una manera automática, remota y visual, aunque también es posible calibrar un sistema fotovoltaico de manera local haciendo uso del botón que hay en la placa. El problema es que este script solo se puede ejecutar si tenemos instalado MATLAB, pero como ya hemos comentado, también tenemos una aplicación que podemos ejecutar en cualquier PC y que cuyo resultado de la ejecución podemos ver a continuación:

Para ejecutar la aplicación implementada en MATLAB solo hay que seleccionar el puerto serie en el que está instalado Arduino y hacer clic en el botón calibrar y el sistema se ajustará automáticamente, y enviará los datos tomados, tal como se muestra en la siguiente figura:

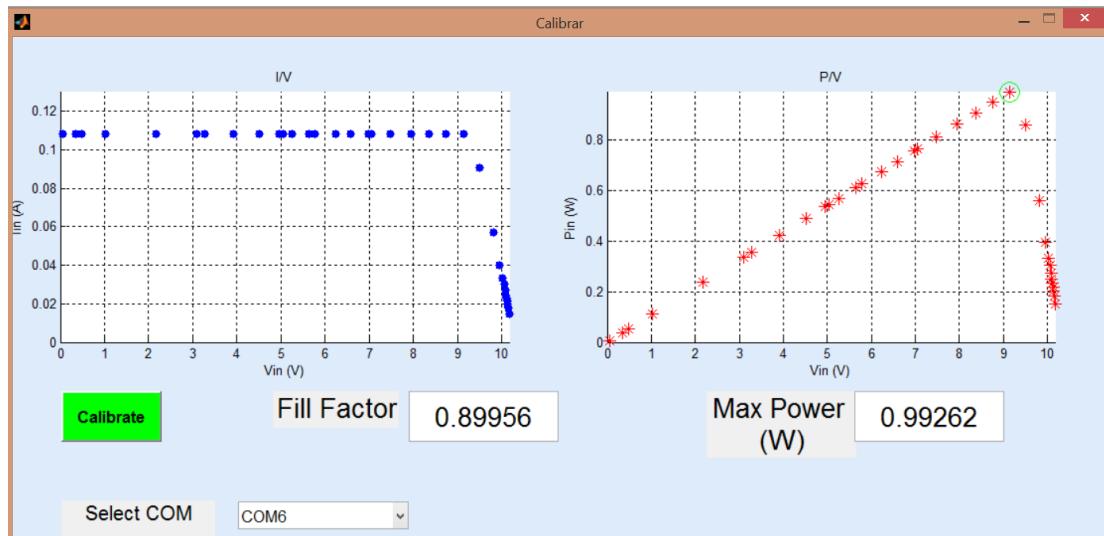
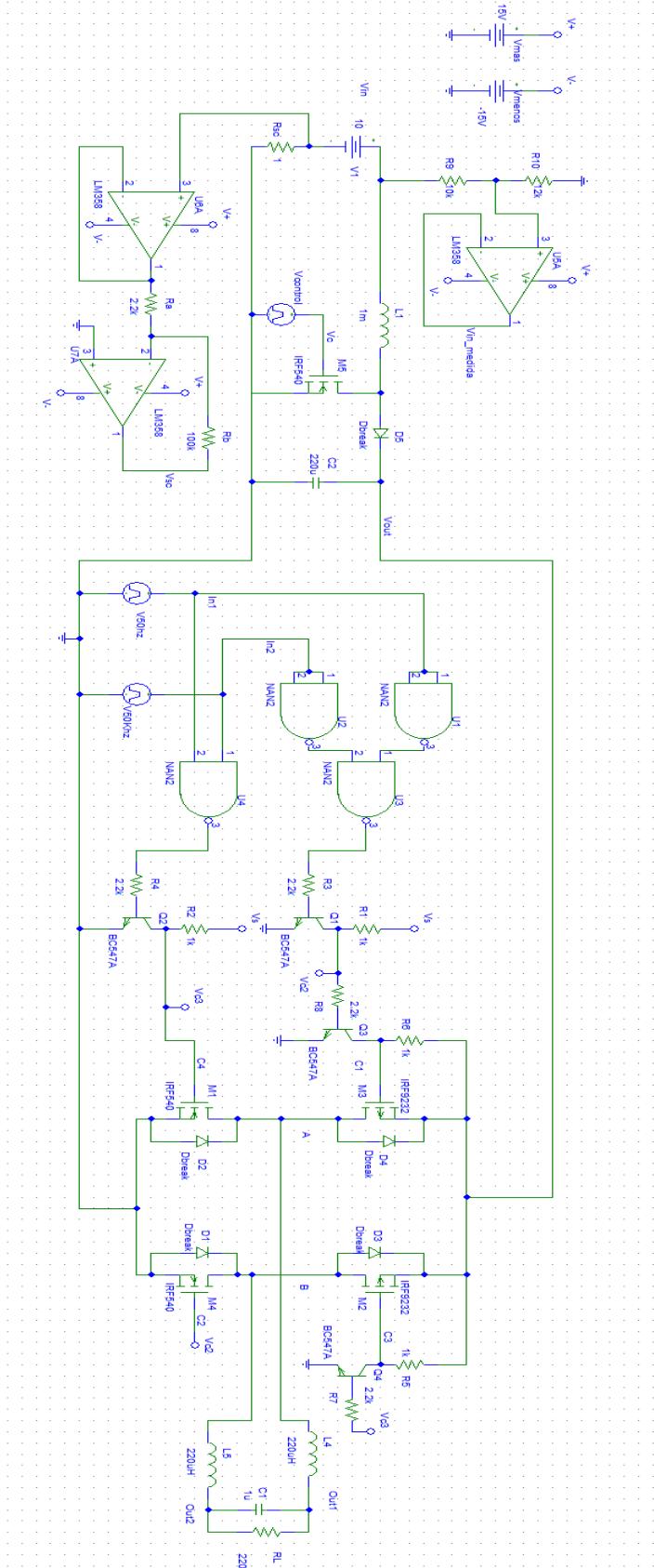


Figura 7.4: Interfaz de usuario operativa.

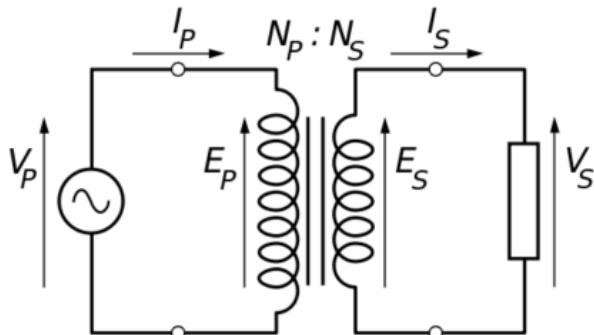
Como podemos ver hemos seleccionado el puerto COM6 que es en el que está Arduino y conforme el sistema va ajustándose, la aplicación nos va mostrando las curvas I-V y P-V, el factor de llenado y la máxima potencia. Si volvemos a calibrar, la aplicación borra los datos y se reinicia. En este caso obtuvimos una potencia de aproximadamente 1W y un factor de llenado de casi el 90%, lo que es una eficiencia bastante elevada.

## **Apéndice I. Esquemático del circuito completo.**



## Apéndice II. Líneas futuras. Transformador.

Como ya comentamos en el capítulo de introducción, un transformador [39] es un dispositivo electrónico que permite aumentar o disminuir la tensión en un circuito eléctrico de corriente alterna, manteniendo la potencia, de manera que la energía se puede transmitir largas distancias sin pérdidas por disipación de calor. A continuación podemos ver el esquema eléctrico de un transformador:



Este elemento se basa en el fenómeno de inducción electromagnética, ya que si aplicamos una fuerza electromotriz alterna en el devanado primario, debido a la intensidad y sentido de la corriente alterna, se produce una inducción de un flujo magnético variable en el núcleo de hierro. Este flujo magnético genera por inducción electromagnética crea una fuerza electromotriz en el devanado secundario. La tensión y la corriente en el secundario dependerán directamente del número de espiras que tengan los devanados y la tensión y la corriente en el primario. A continuación podemos ver cómo se relacionan la tensión y la corriente con el número de espiras de los devanados:

$$\frac{Vp}{Vs} = \frac{Is}{Ip} = \frac{Np}{Ns}$$

## Apéndice III. Líneas futuras. Almacenador.

La función prioritaria de las baterías en un sistema de generación fotovoltaico es la de acumular la energía que se produce durante las horas de luminosidad para poder ser utilizada en la noche o durante períodos prolongados de mal tiempo.

### 8.2.1 Principio de funcionamiento

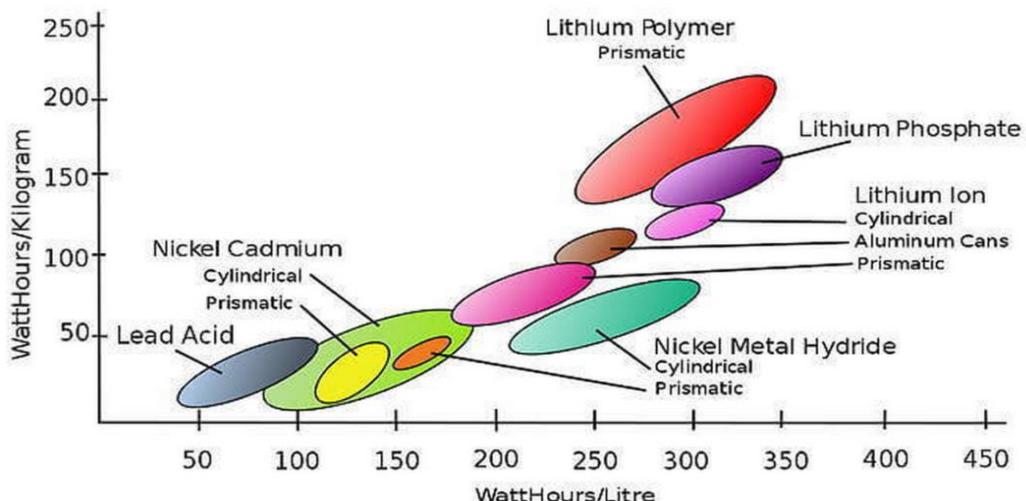
El principio de funcionamiento de un acumulador está basado en un proceso químico reversible llamado reducción –oxidación, en el cual uno de los componentes se oxidan ( pierde electrones) y el otro se reduce (gana electrones), de esta manera los componentes no se consumen sino que cambian su estado de oxidación, lo que les permite volver a su estado original cuando se cierra el circuito al que están conectados y se les aplica una corriente durante la carga.

### 8.2.3 Tipos de baterías

Por lo que a las características electroquímicas se refiere, se encuentran en el mercado diferentes tipos de acumuladores:

- Baterías de plomo-ácido
- Baterías de níquel-hierro
- Baterías de níquel-cadmio
- Baterías de níquel-hidruro metálico

A continuación podemos ver los diferentes tipos de baterías en función de la potencia que generan y su densidad.



Tipos de baterías con su capacidad por volumen y peso. [41]

Entre las baterías disponibles en el mercado, la más idónea para aplicaciones fotovoltaicas resulta ser la batería de plomo, sobre todo gracias al rendimiento de carga y descarga, y a la relación precio prestaciones

## **Apéndice IV. Acrónimos.**

FF: Fill Factor.

CTC: Compare Time on Compare Match.

DC: Direct Current.

AC: Alternating Current.

D: Duty cycle.

UF: Ultra Fast.

ADC: Analog to Digital Converter.

TFG: Trabajo Fin de Grado.

OCR: Output Compare Registers.

ICR: Input Compare Registers.

## Referencias

- [4] [https://es.wikipedia.org/wiki/Unidad\\_de\\_Potencia\\_Neta](https://es.wikipedia.org/wiki/Unidad_de_Potencia_Neta)
- [6] [https://es.wikipedia.org/wiki/Efecto\\_fotoeléctrico](https://es.wikipedia.org/wiki/Efecto_fotoeléctrico)
- [7] [https://es.wikipedia.org/wiki/Panel\\_fotovoltaico](https://es.wikipedia.org/wiki/Panel_fotovoltaico)
- [10] <http://www.dforcesolar.com/energia-solar/mapa-de-radiacion-solar-en-el-mundo/>
- [14] <http://blog.greendok.com/dimensionado-de-una-instalacion-fotovoltaica-condiciones-climaticas-y-zona/>
- [15] <http://ecoinventos.com/los-invernaderos-podrian-generar-la-electricidad-que-consume-andalucia/>
- [17] <http://enersave.es/medicion-neta/energia-solar>
- [18] [http://ocw.unia.es/ciencias-tecnologicas/tecnologia-de-celulas-y-modulos-fotovoltaicos/Materiales/unidad-1/skinless\\_view](http://ocw.unia.es/ciencias-tecnologicas/tecnologia-de-celulas-y-modulos-fotovoltaicos/Materiales/unidad-1/skinless_view)
- [20] [https://es.wikipedia.org/wiki/Convertidor\\_Boost](https://es.wikipedia.org/wiki/Convertidor_Boost)
- [21] [http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Complete.pdf](http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf)
- [22] <http://ww1.microchip.com/downloads/en/DeviceDoc/21422D.pdf>
- [23] [https://es.wikipedia.org/wiki/Divisor\\_de\\_tensión](https://es.wikipedia.org/wiki/Divisor_de_tensión)
- [24] <http://www.electronicafacil.net/tutoriales/SEGUIDOR-DE-TENSION.php>
- [25] <http://www.ti.com/lit/ds/symlink/lm158-n.pdf>
- [26] [https://es.wikipedia.org/wiki/Amplificador\\_operacional](https://es.wikipedia.org/wiki/Amplificador_operacional)
- [28] [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lmt/juarez\\_s\\_eu/capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/juarez_s_eu/capitulo4.pdf)
- [30] [https://en.wikipedia.org/?title=Pulse-width\\_modulation](https://en.wikipedia.org/?title=Pulse-width_modulation)
- [31] <https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CCcQFjAB&url=http%3AA%2Fwww.uv.es%2Femaset%2Fiep00%2Fdescargas%2FIversores%2520PWM%252006-07.ppt&ei=ruOCVbbIGsi9Ud3ygcgE&usg=AFQjCNF3Vil0yct8n-mRQZyBG4W2wXo10w&sig2=PdUYDZhIJi9xaH68K8aHPg&bvm=bv.96041959,d.d24&cad=rja>
- [32] [http://www.datasheetcatalog.com/info\\_redirect/datasheets/50/383972\\_DS.pdf.shtml](http://www.datasheetcatalog.com/info_redirect/datasheets/50/383972_DS.pdf.shtml)
- [35] <http://www.monografias.com/trabajos37/entrada-y-salida/entrada-y-salida.shtml>
- [36] <http://4cp2ok2012g9.blogspot.com.es/2012/06/tp-n4.html>
- [37] <http://perso.wanadoo.es/pictob/comserie.htm>
- [40] <https://es.wikipedia.org/wiki/Transformador>
- [41] [https://es.wikipedia.org/wiki/Batería\\_recargable](https://es.wikipedia.org/wiki/Batería_recargable)

A continuación tenemos más información sobre los temas que hemos ido tratando en cada capítulo del proyecto.

# Bibliografía

## Enlaces de interés

<http://www.arduino.cc/>  
<http://es.mathworks.com/products/matlab/>  
<http://www.datasheetcatalog.com/>

## 2. Energía solar fotovoltaica

Escala de Kardashov:

[1][https://es.wikipedia.org/wiki/Escala\\_de\\_Kardashov](https://es.wikipedia.org/wiki/Escala_de_Kardashov)

Energía solar:

[2]<https://es.wikipedia.org/wiki/Sol>

Unión PN:

[3]<http://www.ifent.org/lecciones/diodo/default.asp>

Efecto fotoeléctrico:

[5]<https://books.google.es/books?id=wo9tYi5WS0MC&pg=PA93&hl=es#v=onepage&q&f=false>

Efecto Joule:

[8][https://es.wikipedia.org/wiki/Efecto\\_Joule](https://es.wikipedia.org/wiki/Efecto_Joule)

[8]<http://hipertextual.com/2015/05/efecto-joule>

Radiación solar:

[9][https://es.wikipedia.org/wiki/Energ%C3%ADa\\_solar\\_fotovoltaica](https://es.wikipedia.org/wiki/Energ%C3%ADa_solar_fotovoltaica)

## 3. Caracterización de un panel fotovoltaico

[CAR] Miguel Ángel Carvajal. Práctica de Sistemas de alimentación.

[16][https://es.wikipedia.org/wiki/Panel\\_fotovoltaico](https://es.wikipedia.org/wiki/Panel_fotovoltaico)

## 4. Desarrollo del convertidor elevador o Boost

[CAR] Miguel Ángel Carvajal. Transparencias de sistemas de alimentación

[19][https://books.google.es/books?id=aK\\_Eji4biusC&pg=PA255&dq=convertidor+boost&hl=es&sa=X&ei=au-CVeTfPImnU7rxkqAM&ved=0CFIQ6AEwBQ#v=onepage&q=convertidor%20boost&f=false](https://books.google.es/books?id=aK_Eji4biusC&pg=PA255&dq=convertidor+boost&hl=es&sa=X&ei=au-CVeTfPImnU7rxkqAM&ved=0CFIQ6AEwBQ#v=onepage&q=convertidor%20boost&f=false)

## 5. Desarrollo del inversor

[27][https://books.google.es/books?id=aK\\_Eji4biusC&pg=PA87&dq=inversor+puente+completo&hl=es&sa=X&ei=6kKIVbm2EqnRygO\\_3joAQ&ved=0CEUQ6AEwBg#v=onepage&q=inversor%20puente%20completo&f=false](https://books.google.es/books?id=aK_Eji4biusC&pg=PA87&dq=inversor+puente+completo&hl=es&sa=X&ei=6kKIVbm2EqnRygO_3joAQ&ved=0CEUQ6AEwBg#v=onepage&q=inversor%20puente%20completo&f=false)

PWM:

[29]<https://books.google.es/books?id=8LGi1AjSfpC&printsec=frontcover&dq=PWM&hl=es&sa=X&ei=wUCIVfbRMoL5ygOb8IB4&ved=0CCAQ6AEwAA#v=onepage&q=PWM&f=false>

Circuito LC:

[33]<https://books.google.es/books?id=SghjkM6MwygC&pg=PA877&dq=circuito+lc&hl=es&sa=X&ei=tKCJVZODGaqqywOMwa-wAQ&ved=0CCcQ6AEwAQ#v=onepage&q=circuito%20lc&f=false>

## 6. Comunicación Arduino-PC

[34]<http://perso.wanadoo.es/pictob/comserie.htm>

## 8. Interfaz de usuario

[38]<https://www.youtube.com/watch?v=tJGbqaC-PU0>

[38]<http://es.mathworks.com/videos/creating-a-gui-with-guide-68979.html>

