

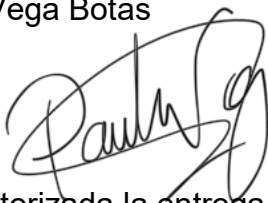
Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título:

*DISEÑO Y FABRICACIÓN DE UN SISTEMA DE CONTROL DE
EMISIÓN DE LUZ DE UN LEC Y SU MONITORIZACIÓN*

.....
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico2018-2019.... es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.

Fdo.: Paula De Vega Botas

Fecha: 26.../ 08.../ ...2019



Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Dr. José Sánchez del Río Sáez

Fecha:...26/08/2019



AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. **PAULA DE VEGA BOTAS**

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: **DISEÑO Y FABRICACIÓN DE UN SISTEMA DE CONTROL DE EMISIÓN DE LUZ Y SU MONITORIZACIÓN. ADAPTACIÓN DE DICHA SEÑAL A LA PLACA DE UN SENSOR DE IMPEDANCIA**, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducir la en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- Madrid, a **24 de agosto de 2019**

Fdo **PAULA DE VEGA BOTAS**



Autorizada la entrega del proyecto del alumno:

Paula de Vega Botas

EL DIRECTOR DEL PROYECTO

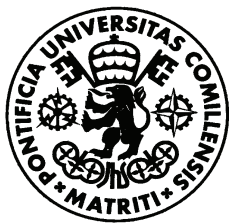
José Sánchez del Río Sáez

Fdo.: Fecha: / /

Vº Bº DEL COORDINADOR DE PROYECTOS

Francisco Javier Herraiz Martínez

Fdo.: Fecha: / /



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN

TRABAJO DE FIN DE GRADO

**DISEÑO Y FABRICACIÓN DE UN SISTEMA DE
CONTROL DE EMISIÓN DE LUZ DE UN LEC Y SU
MONITORIZACIÓN**

AUTOR: Paula de Vega Botas

DIRECTOR: José Sánchez del Río Sáez

MADRID, Agosto de 2019

*A mis padres,
Estrella y Miguel Ángel.*

*Ninguna invención
es perfecta al nacer.*
CICERÓN

*No estoy desanimado porque
cada intento equivocado
descartado es un paso adelante.*
THOMAS A. EDISON

Agradecimientos

En primer lugar, debo dar las gracias a mi profesor de electrónica y circuitos, José Luis Rodríguez Marrero. Sin duda sus asignaturas y su modo de impartirlas me han proporcionado una capacidad de análisis y comprensión que he empleado para el diseño de este sistema. También agradezco que José Luis haya seguido de cerca el avance del circuito y agradezco haber podido acudir a él con todas las dudas que me ha generado el diseño del mismo.

Doy las gracias también a mi director de TFG, José Sánchez del Río por ayudarme con el proyecto cuando me he atascado, ayudarme a utilizar el material que había en el laboratorio y por suministrarme todos los componentes que he necesitado durante el periodo que he estado yendo a IMDEA Materiales. Gracias a él y los recursos que había en el laboratorio que me ha mostrado en momentos críticos he podido sacar en adelante el proyecto. También doy gracias a IMDEA Materiales por permitirme usar sus instalaciones y equipos durante el desarrollo de mi trabajo de fin de grado.

También debo dar gracias a mis compañeros, que de una manera o de otra han aportado un pequeño grano de arena a este trabajo. De apoyarnos mutuamente durante todas las épocas de exámenes que hemos pasado a intentar resolver fallos en el código de otra persona o encontrar el error de un circuito. Ana B., Marina, Conchi, Ana H., Gimena, Sergio, Dani, Jose y Juan: creo de verdad que sin el apoyo que nos hemos brindado los unos a los otros y la colaboración constante no habríamos llegado hasta este punto tan satisfechos y felices. Espero que vosotros hayáis sentido mi apoyo tanto como yo siento el vuestro.

Por último, pero no por ello menos importante, me gustaría dar las gracias a mis padres y a mi hermana. Habéis sido mi guía y apoyo en momentos muy difíciles. Nuria, de ti he aprendido que aunque haya obstáculos en el camino si uno quiere, los podrá superar y seguir adelante. Eso es una lección que podré aplicarme siempre en la vida, acordándome de ti. Papá, mamá, siempre os estaré agradecida de haberme inculcado valores como el respeto, tolerancia y gratitud. Para lo que a este trabajo concierne, la pulcritud, el esmero, la perseverancia y la dedicación que espero demostrar en este documento y en todo lo que hago. Sin vuestro apoyo incondicional no podría haber llegado hasta donde estoy a día de hoy.

Resumen

DISEÑO Y FABRICACIÓN DE UN SISTEMA DE CONTROL DE EMISIÓN DE LUZ DE UN LEC Y SU MONITORIZACIÓN

Autor: Paula de Vega Botas.

Director: José Sánchez del Río Sáez.

Abstract.- Los cristales LEC (*light-emitting electrochemical cell*) son un invento revolucionario y una opción que puede desbancar a las lámparas incandescentes, fluorescentes y LED como fuente de iluminación, pues son alternativas más eficientes y de bajo coste. Estos cristales están aún bajo desarrollo y parte de la investigación de los mismos consiste en el estudio de su comportamiento y el comportamiento de su impedancia al inyectar señales pulsadas de 0 V a 25 V de voltaje y de 0 mA a 100 mA de corriente. Estas señales que se desean inyectar en los cristales LEC deben tener un barrido en frecuencias de 0.1 Hz hasta frecuencias del orden de MHz. Además, la impedancia se debe medir en las zonas bajas de la señal de voltaje, por lo que el sistema a desarrollar debe quedar acoplado a un sistema medidor de impedancia y enviar un *flag* de *trigger* cada vez que el voltaje esté a nivel bajo. En este trabajo se ha diseñado un sistema de pulsado de luz y fuente de corriente automatizado y controlado por una interfaz de usuario. Debido a la falta de tiempo, el trabajo se acotó para trabajar en frecuencias de 5 Hz a kHz y se obvió la parte de envío de la señal de *trigger* al correspondiente medidor de impedancia en niveles bajos de la señal.

Palabras clave: LEC, voltaje, corriente, potencia, programa, amplificación, ganancia.

Resumen del proyecto.

1. Introducción.

Un cristal LEC es un material multicapa que contiene una capa activa con un semiconductor electroluminiscente que emite luz visible al aplicarse una diferencia de voltaje entre sus electrodos gracias a un fenómeno llamado dopaje. El dopaje consiste en dos zonas, una zona p- y una zona n- separadas por una región no dopada. Al aplicar una diferencia de voltaje las zonas dopadas comienzan a crecer y a emitir luz y la zona no dopada se va reduciendo hasta que las zonas dopadas se encuentran y es entonces cuando el LEC deja de emitir luz. Los cristales LEC fueron desarrollados por Pei, Q et al. en 1995. [1],[2]. El desarrollo de los cristales LEC tiene una parte de investigación del comportamiento de la impedancia de los mismos. La impedancia de los LEC se mide inyectando al LEC una señal de voltaje variable de 0 V a 25 V y midiendo la fase que hay entre este voltaje y la corriente que fluye por el LEC.

2. Descripción de proyecto.

El proyecto va a consistir en el diseño de un sistema que suministre al cristal LEC una tensión pulsada variable de 0V a 25 V de amplitud y una corriente variable que fluya por el LEC y que pueda variar de 0 mA a 100 mA. Esta señal, así como el tipo de onda que debe

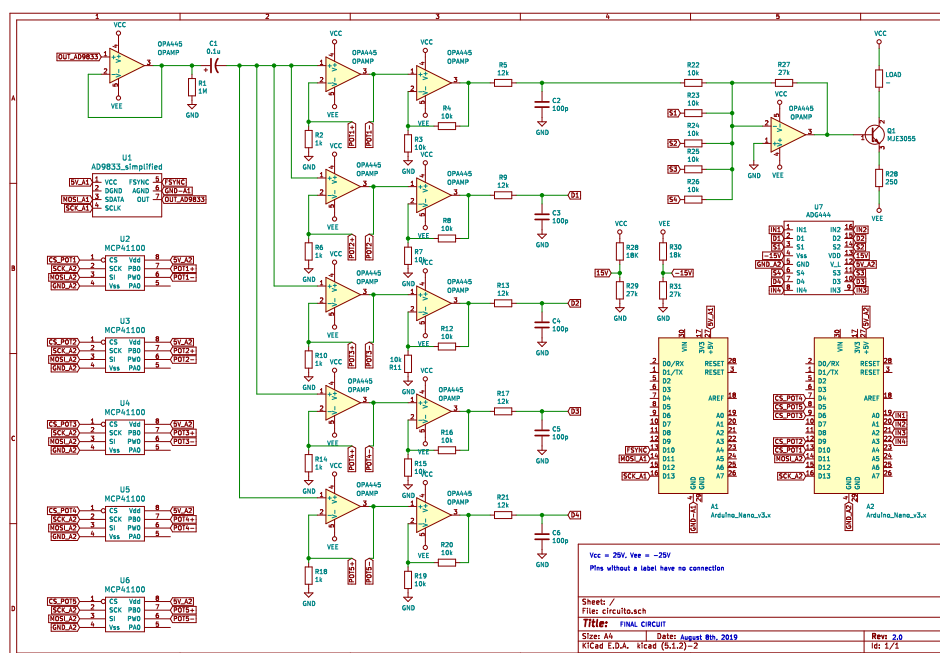
ser preferiblemente una señal sinusoidal y la corriente suministrada al LEC debe poder ser controlado desde una interfaz gráfica amigable para el usuario final. Debe ser un proceso automatizado.

3. Materiales y métodos.

El proyecto ha tenido varias iteraciones y varios diseños del circuito se han descartado porque no cumplían con la funcionalidad pedida. Para conseguir la potencia deseada de la señal a inyectar al LEC primero hay que generar la misma señal. Esta tarea se consigue con la tarjeta AD9833 que queda controlada por un Arduino. Una vez generada la señal, será tratada por varias etapas de filtrado, amplificación y suma para conseguir la potencia deseada en la señal. El circuito tiene 5 subcircuitos iguales y se puede dividir en varias etapas:

- Filtro paso alto. Para eliminar el offset que genera la tarjeta AD9833.
- Etapa de amplificación variable. Diseñada con potenciómetros digitales MCP41100 controlados por Arduinos. En función de qué corriente pida el usuario, habrá un valor diferente de ganancia en esta etapa.
- Etapa de amplificación fija. Para aumentar la ganancia total de circuito ya que con una sola etapa de amplificación no se conseguía llegar a los requisitos pedidos. [3],[4].
- Filtro paso bajo. El reloj de 10 MHz del potenciómetro digital utilizado en la etapa de amplificación variable interfiere con el circuito e inyecta ruido de altas frecuencias. Se incluye una etapa de filtrado para eliminar este ruido.
- Etapa de suma. Las señales de cada uno de los 5 subcircuitos se suman para mandarlas a la base de un transistor.
- Transistor como fuente de corriente. Se utiliza un transistor NPN de potencia modelado como fuente de corriente variable controlada por una interfaz de usuario.

A continuación se adjunta un diagrama del sistema en cuestión:



4. Resultados y discusión.

Los resultados del sistema se van a medir en términos de corriente y en términos de voltaje. La corriente se midió utilizando el multímetro Agilent 34410 y colocándolo en serie con el circuito. Desde la interfaz gráfica se fueron insertando valores de corriente deseada desde 5 mA a 100 mA con una variación de 0.5 mA y se recogieron los datos en un Excel. La siguiente gráfica compara la corriente teórica deseada y la corriente que suministra el circuito, así como una recta de aproximación de la corriente que suministra el circuito.

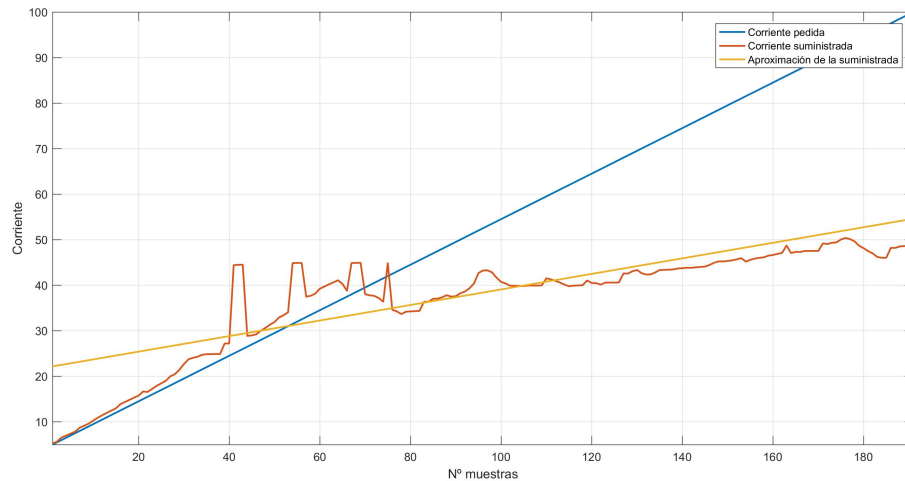


Figure 1. Corrientes a la salida: pedida vs. suministrada y aproximaciones

Se observan varios fenómenos: principalmente, unas anomalías en la corriente suministrada. Si se separan los resultados en antes de las anomalías y después de las anomalías se puede concluir que la corriente suministrada antes de las anomalías se ajusta muy bien a los valores de corriente pedida por el usuario. Sin embargo la corriente suministrada después de las anomalías tiene un rendimiento bastante pobre. En concreto:

$$\eta = \frac{\overline{corrienteSuministrada_{aprox}}}{corrientePedida} = 64.48\%$$

En cuanto al rendimiento en términos de voltaje, la señal de salida tuvo que ser procesada a través de un divisor de tensión para poder ver la señal completa en el osciloscopio. Después se calculó con MATLAB cuál debía ser la señal teórica a la salida y se compararon. La captura del osciloscopio y la comparación de las señales real y teórica se adjuntan a continuación.

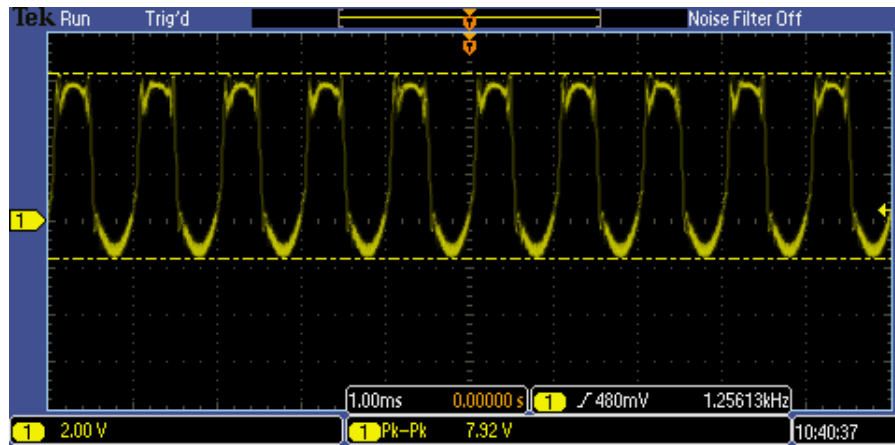


Figure 2. Señal a la salida del divisor auxiliar

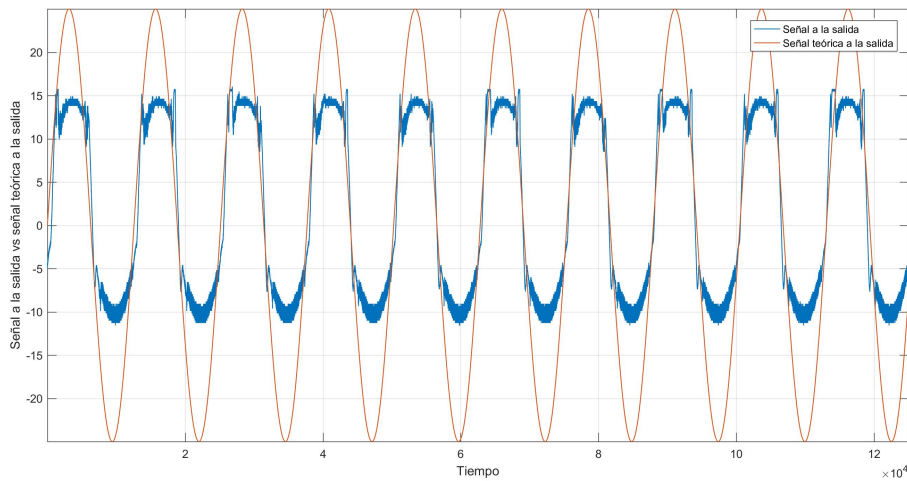


Figure 3. Señales a la salida: real vs. teórica

Calculamos el rendimiento como el cociente entre los voltajes pico-pico de las dos señales:

$$\eta = \frac{V_{pp,señalreal}}{V_{pp,señalteórica}} = \frac{22.0242 - (-5.5758)}{25 - (-25)} = \frac{27.6}{50} = 55.2\%$$

Se puede ver claramente que el sistema no tiene un rendimiento perfecto y que tiene varias limitaciones que se enunciarán a continuación.

5. Limitaciones.

La limitación inicial ha sido la acotación del alcance del sistema debido a falta de tiempo. En lugar de trabajar con barridos de frecuencias de 0.1 Hz hasta MHz se redujo el barrido de 5 Hz hasta kHz. El sistema muestra ciertos fallos debido a que existen ciertos componentes en el circuito, como es el potenciómetro MCP41100 que no se tiene la certeza que pueda suministrar altas potencias. Esto ha generado incógnitas en el circuito que no se ha tenido tiempo a resolver. Además, este potenciómetro también ha generado un nivel medio en el voltaje de salida que no interesa para el estudio de la impedancia de los cristales LEC.

6. Referencias.

[1].Pei Q, Y. G. *Polymer light-emitting electrochemical cells*. 1995.

- [2]. **Ruben D. Costa, E. O.** *Luminescent Ionic Transition-Metal Complexes for Light-Emitting Electrochemical Cells*. *Angewandte Reviews*. 2012.
- [3]. **Texas Instruments.** *Amplificador operacional OPA445*. Retrieved from: <https://www.ti.com/lit/ds/symlink/opa445.pdf>
- [4]. **Paul Horowitz.** *The Art Of Electronics - 2nd Edition*. 1989.

Summary

DISEÑO Y FABRICACIÓN DE UN SISTEMA DE CONTROL DE EMISIÓN DE LUZ DE UN LEC Y SU MONITORIZACIÓN

Author: Paula de Vega Botas.

Director: José Sánchez del Río Sáez.

Abstract.- LEC (light-emitting electrochemical cell) crystals are a revolutionary invention that could potentially knock incandescent, fluorescent and LED lamps off as light sources, because they are cheaper, low cost and more efficient alternatives. These crystals are still under development and part of the research consists on studying the behaviour of the crystals themselves and the behaviour of their impedance when 0 V to 25 V voltage pulse signals and 0 mA to 100 mA current pulse signals are flowing through the crystal. These signals that will be injected into the LEC crystal must have a frequency sweep starting at 0 Hz and frequencies up to MHz. Moreover, the impedance of the crystal must be measured in the low stages of the voltage signal, thus the system to be developed has to be coupled to an impedance measuring system and must send a trigger flag to this impedance measuring system everytime the voltage signal is at its low stage. This project consists of the design of a system which will emit pulsed light and will work as an automated current source controlled by an user interface. Due to lack of tim, this project has been bounded in order to work with frequency sweeps from 5 Hz to kHz and the part of generating and sending a trigger flag to the corresponding impedance measuring system was skipped.

Keywords: LEC, voltage, current, power, program, amplification, gain.

Summary of the project.

1. Introduction.

A LEC crystal is a multi-layer material that contains an active layer made of an electroluminescent semiconductor that emits light when a bias voltage is applied to the electrodes of the crystal thanks to a phenomenom called doping. This doping phenomenom consists of two doped regions; a p- region and a n- region separated by a non-doped region or neutral region. When a bias voltaje is applied the doped regions start to increase in size and emit light. The neutral region starts decreasing in size until both doped regions meet. At this poing the LEC crystal will have stopped to emit light. LEC crystals where developed by Pei, Q et al. in 1995. [1],[2]. The development of these LEC crystals has a vast research about the behaviour of the impedance of the LECs. LEC impedance is measured by injecting into the LEC a variable voltaje signal with an amplitude that varies between 0 V and 25 V and measuring the phase between the voltage signal and the current that flows through the LEC.

2. Project description.

This project will consist on the design of a system that will supply a pulse-like voltage

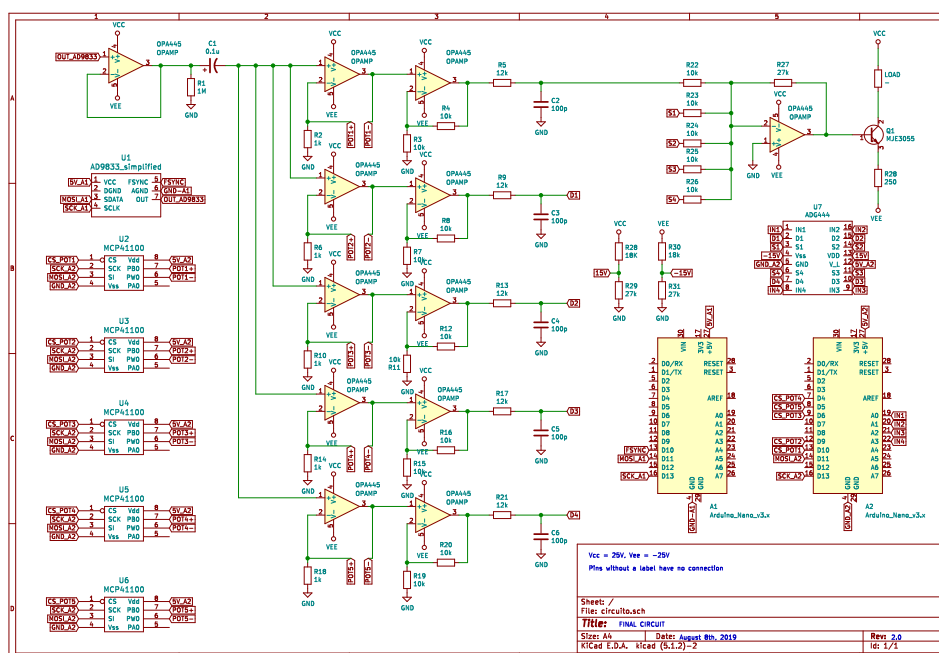
signal whose amplitude will vary from 0 V to 25 V and a variable current that must flow through the LEC crystal and that can vary from 0 mA to 100 mA. This signal, the waveform type (preferably a sine wave) and the current supplied to the LEC must be controlled from a friendly graphical user interface for the final user. This all must be an automated process.

3. Materials and Methods.

The project has had several iterations and various designs for the circuit have been dismissed because they did not comply with the functionality desired for the system. In order to get the desired power of the signal that will be injected to the LEC, first of all, the signal must be generated. This is managed using the AD9833 board that is controlled by an Arduino board. Once the signal is generated it will be processed through several filtering, amplifying and summing stages with the aim of getting the desired power of the signal. The circuit has 5 equal subcircuits and can be divided in the following stages:

- Highpass filter. To eliminate the offset that the AD9833 board generates
- Variable gain stage. Designed with MCP41100 digital potentiometers controlled by Arduino boards. Depending on what current the user needs, there will be a different gain value in this stage.
- Fixed gain stage. To increase the total gain of the circuit. Using only one amplifying stage was not enough to reach the requirements of the project.[3],[4].
- Lowpass filter. The 10 MHz clock from the digital potentiometer used in the variable gain stage creates interferences with the circuit and injects high frequency noise into the circuit. This filter is included in order to eliminate said high frequency noise.
- Signal summing stage. Each subcircuit generates a signal and all of them must be summed up into a greater signal that will go into the base of a transistor.
- Transistor modeled as current source. A power NPN transistor is used to model a variable current source controlled by an user interface.

The following figure represents a diagram of the previously mentioned system:



4. Results and discussion.

The results of this system are going to be measured in terms of current and in terms of voltage. Current was measured using the Agilent 34410 multimeter and coupling it in series with the rest of the circuit. From the user interface different desired current values were inserted, from 5 mA to 100 mA with a 0.5 mA step. These data was collected in an Excel spreadsheet. The following graph compares the theoretical desired current and the current supplied by the circuit, as well as an approximation function to the current that the circuit supplies.

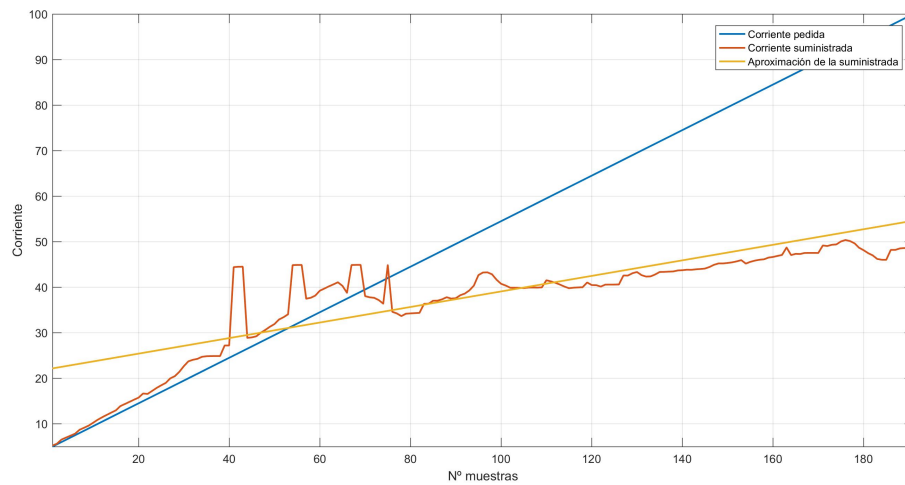


Figure 4. Output currents: desired vs. supplied and approximations

Several phenomena can be observed in this graph, but mainly some anomalies in the current supplied by the circuit. The results can be split in two parts: results before anomalies and after anomalies. The current supplied by the circuit before anomalies occurred is very well adjusted to the current desired by the user. Nevertheless, the supplied current after anomalies occurred has a rather poor performance. Specifically:

$$\eta = \frac{\text{suppliedCurrent}_{\text{aprox}}}{\text{desiredCurrent}} = 64.48\%$$

For the measurement of performance in terms of voltage the output signal had to be processed through a voltage divider in order to see the complete signal in the oscilloscope screen. After this, some calculations were made with MATLAB to find out what the theoretical output signal should be and this theoretical signal and the real one were compared. The following figures show the oscilloscope screenshot and the MATLAB comparison between both signals.

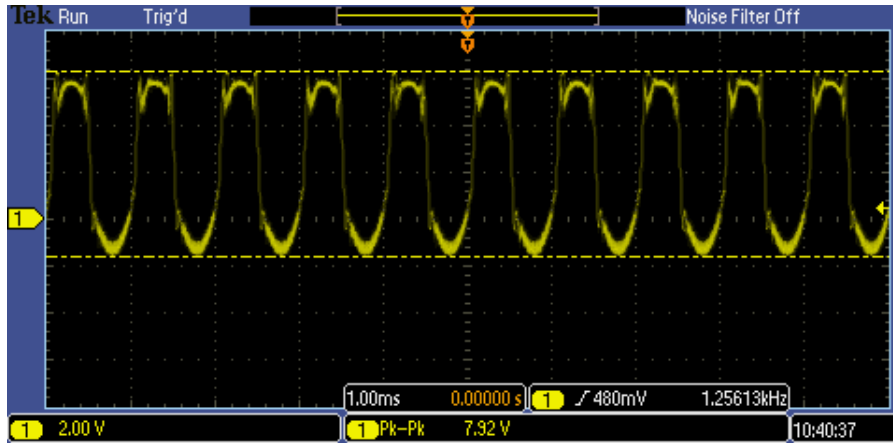


Figure 5. Output signal after a voltage divider

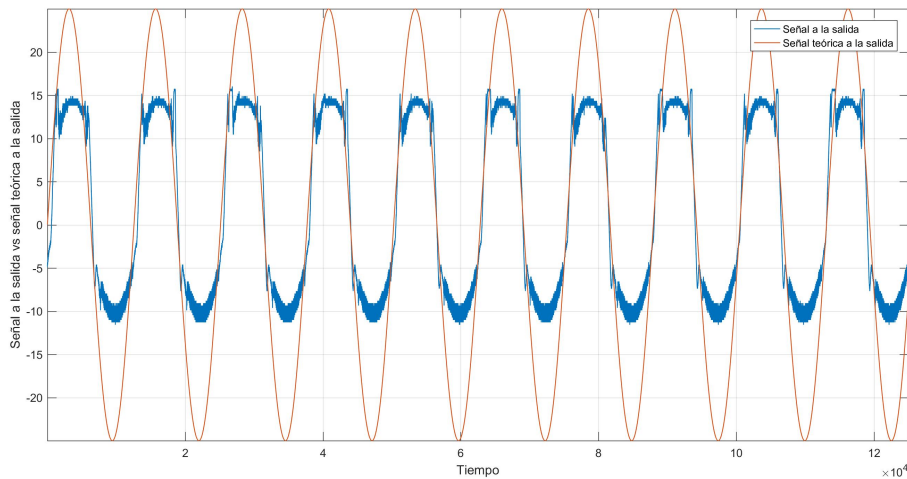


Figure 6. Output signals: real vs. theoretical

Performance in terms of voltage can be measured as the ratio between the peak-to-peak voltages of both signals mentioned previously:

$$\eta = \frac{V_{pp,real\,signal}}{V_{pp,theoretical\,signal}} = \frac{22.0242 - (-5.5758)}{25 - (-25)} = \frac{27.6}{50} = 55.2\%$$

It is very clear that the system does not have a perfect performance and that it has several limitations that will be described in the next section.

5. Limitations.

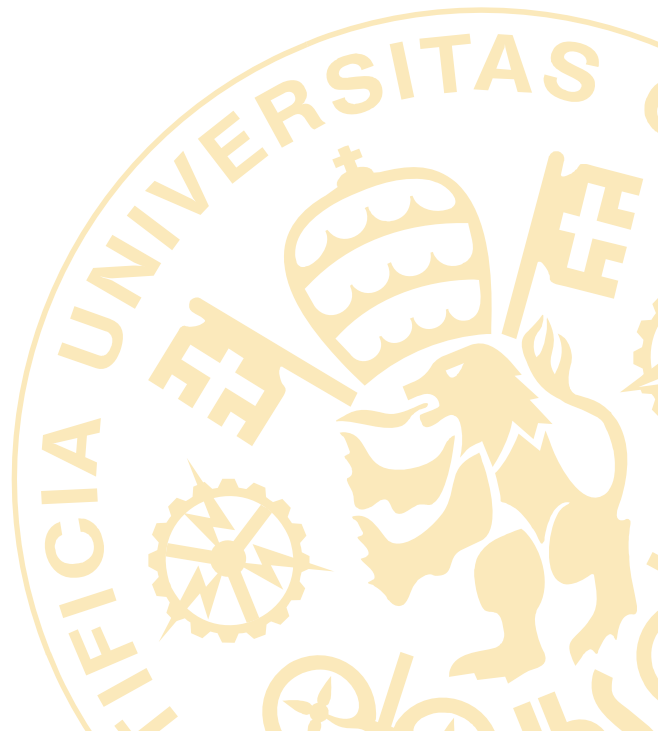
The initial limitation of this project has been the reduction of the scope of the system due to lack of time. Instead of doing a frequency sweep from 0.1 Hz to MHz the frequency sweep was reduced to 5 Hz up to kHz. The system shows various errors due to the presence of components in the circuit that there is no proof can supply enough power, as the MCP41100 potentiometer. This, among other things, has generated uncertainties in the circuit that there has not been enough time to research and solve. Furthermore, this potentiometer has also generated an offset in the output voltage that is not desirable for the study of the behaviour of the impedance of the LEC crystals.

6. References.

- [1]. **Pei Q, Y. G.** *Polymer light-emitting electrochemical cells*. 1995.
- [2]. **Ruben D. Costa, E. O.** *Luminescent Ionic Transition-Metal Complexes for Light-Emitting Electrochemical Cells*. *Angewandte Reviews*. 2012.
- [3]. **Texas Instruments.** *Amplificador operacional OPA445*. Retrieved from: <https://www.ti.com/lit/ds/symlink/opa445.pdf>
- [4]. **Paul Horowitz.** *The Art Of Electronics - 2nd Edition*. 1989.

DOCUMENTO I

MEMORIA



Índice

I. Memoria	11
1. Introducción	13
1.1. Motivación	13
1.2. Objetivos	14
1.3. Recursos	15
1.4. Metodología	22
1.5. Organización del documento	24
2. Estado del Arte	25
2.1. Introducción	25
2.2. Qué es un LEC	26
2.3. Turn-on time y lifetime del LEC	26
2.3.1. Lifetime o tiempo de vida del LEC	26
2.3.2. Tiempo de encendido o device turn-on time	26
2.4. Formación de la unión p-n	27
2.5. Arquitectura del dispositivo LEC	27
2.6. Materiales de los electrodos	28
2.7. El cristal LEC de IMDEA Materiales	28
2.8. Característica V-R del LEC	29
3. Descripción del Sistema	31
3.1. Generación de la señal con el AD9833	31
3.1.1. Imágenes	32
3.2. El circuito	34
3.2.1. Diagrama de bloques del subcircuito: filtrado y amplificación	35
3.2.2. Etapa 1: Filtrado	35
3.2.3. Etapa 2: Amplificación	36
3.2.3.1. El potenciómetro digital MCP41100	37
3.2.3.2. Funcionamiento de MCP41100	37
3.2.3.3. Diagrama de la primera etapa amplificadora	38
3.2.4. Etapa 3: Amplificación	39
3.2.5. Etapa 4: Filtrado	39
3.2.6. Etapa 5: Suma	40
3.2.7. Etapa 6: Transistor como fuente de corriente	42
3.2.7.1. Motivo de esta última etapa	42
3.2.7.2. Problemas encontrados durante la implementación de esta etapa	42
3.2.7.2.1. Elección del transistor	42
3.2.7.2.2. Elección de la resistencia de emisor	43
3.3. Funcionamiento del sistema	43
3.3.1. La interfaz del usuario	43
3.3.1.1. Generación de comandos para enviar a los Arduinos	44

3.3.1.2.	El comando del circuito	45
3.3.1.3.	Función de mapeo corriente-resistencia	46
3.3.1.4.	Funcionamiento del circuito	48
3.4.	Envío de comandos por serial desde MATLAB	51
4.	Integración del Sistema	53
4.1.	Soldadura del circuito	53
4.2.	Integración en una caja	54
5.	Resultados	59
5.1.	Resultados del sistema	59
5.1.1.	Voltaje a la salida del circuito	59
5.1.1.1.	Rendimiento en términos de voltaje	61
5.1.2.	Corriente pedida vs. Corriente suministrada	61
5.1.2.1.	Rendimiento en términos de corriente	66
5.2.	Problemas encontrados	67
5.2.1.	Anomalías del potenciómetro MCP41100	67
5.2.1.1.	Anomalías en resistencia	67
5.2.1.2.	<i>Soft Clipping</i> o recorte de la señal	67
5.2.1.3.	Ruido de alta frecuencia	69
5.2.2.	Nivel medio	69
5.3.	Soluciones	69
5.3.1.	Ruido de alta frecuencia: filtro paso bajo	69
5.3.2.	¿Solución al <i>Soft Clipping</i> ?	70
6.	Conclusiones y futuros desarrollos	73
6.1.	Conclusiones	73
6.2.	Mejoras y futuros desarrollos	74
6.3.	Análisis del circuito	75
	Bibliografía	77
II.	Código fuente	79
1.	Arduino	81
1.1.	Código de programación del AD9833	81
1.2.	Código de programación potenciómetros y switches	84
2.	Matlab	89
2.1.	Función de mapeo corriente a resistencia	89
2.2.	Función de generación de comando a enviar al Arduino	91
2.3.	Código de la aplicación de interfaz gráfica	93
III.	Estudio económico	97
1.	Estudio económico	99
1.1.	Componentes electrónicos	99
1.2.	Licencias de software	99
1.3.	Suma total	100
IV.	Manual de usuario	101

List of Figures

1. Arduino NANO	15
2. AD9833	16
3. Multímetro Agilent	16
4. Potenciómetro MicroChip MCP41100 100 k Ω	17
5. Amplificador operacional OPA445AP	17
6. Transistor MJE305	18
7. Switch analógico ADG444	18
8. Potenciómetro de potencia	18
9. Potenciómetro de potencia	19
10. Resistencia de potencia	19
11. Resistencias	19
12. Condensador electrolítico de 0.1 μ F	20
13. Condensador de 100pF	20
14. Pines header macho	20
15. Cables dupont hembra-hembra	21
16. Clema	21
17. Placa para soldar el circuito	21
18. Soldador y estaño	22
19. LEC que emite un mensaje	26
20. Formación del dopaje p-n	27
21. Célula sándwich y célula de superficie	28
22. LEC desarrollado en IMDEA	28
23. LEC desarrollado en IMDEA	28
24. Característica V-R del LEC	29
25. Señal de frecuencia 1 kHz generada por el AD9833	32
26. Señal de frecuencia 2.5 kHz generada por el AD9833	33
27. Señal de frecuencia 10 kHz generada por el AD9833	33
28. Señal de frecuencia 25 kHz generada por el AD9833	33
29. Señal de frecuencia 50 kHz generada por el AD9833	33
30. Señal de frecuencia 100 kHz generada por el AD9833	34
31. Señal de frecuencia 500 kHz generada por el AD9833	34
32. Señal de frecuencia 1 MHz generada por el AD9833	34
33. Red paso alto	36
34. Patillaje del potenciómetro MCP41XXX	37
35. Fórmulas para el cálculo de resistencias entre los terminales	37
36. Etapa amplificadora nº 1	38
37. Etapa amplificadora nº 2	39
38. Red paso bajo	39
39. Circuito sumador	40
40. Circuito sumador. Resistencias nominales	41

41. Circuito sumador con interruptores	42
42. Circuito soldado	53
43. Caja sin montar: tapa con las conexiones	55
44. Interior de la caja (sin terminar)	56
45. Interior de la caja terminada	56
46. Caja cableada y lista para usar	57
47. Divisor de voltaje para calcular el voltaje real a la salida	60
48. Señal a la salida del divisor auxiliar	60
49. Señales a la salida: real vs. teórica	61
50. Corrientes a la salida: pedida vs. suministrada	66
51. Corrientes a la salida: pedida vs. suministrada y aproximaciones	66
52. Soft Clipping a $R_{pot} = 5k\Omega$	68
53. Soft Clipping a $R_{pot} = 23.9k\Omega$	68
54. Circuito filtro paso bajo	70
55. Soft clipping solucionado para $R_{pot} = 5k\Omega$	70
56. Soft clipping solucionado para $R_{pot} = 23.9k\Omega$	70
57. Diseño planteado	74

List of Tables

1. Resistencia vs. Corriente	46
2. Corriente pedida vs. Corriente suministrada	66
3. Tabla de precios de componentes	99
4. Tabla de precios de software	99
5. Tabla de suma total de costes	100

Índice de extractos de código

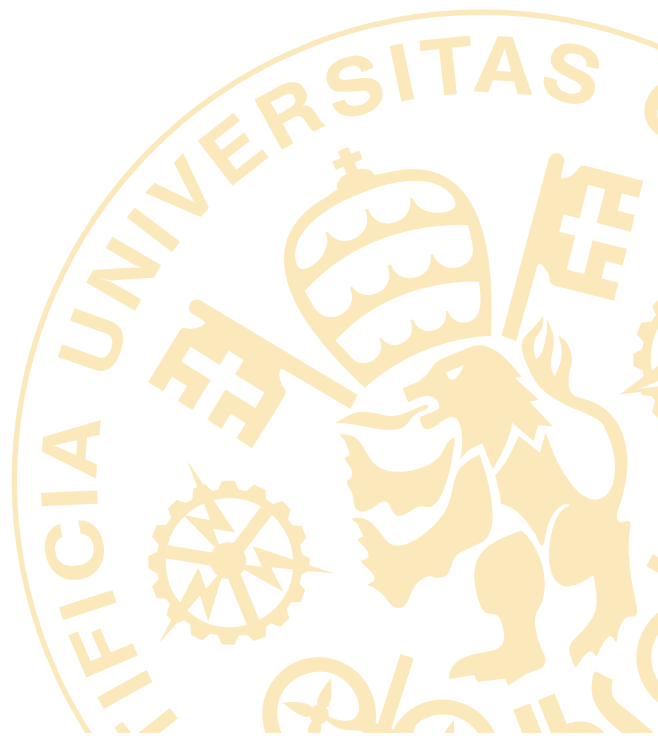
1. Mapeo corriente-resistencia	48
2. Ejemplo de funcionamiento de un sólo subcircuito	49
3. Ejemplo de funcionamiento de tres subcircuitos	49
4. Comando	50
5. Envío de comandos por interfaz serial	51

Acrónimos

<i>TFG</i>	Trabajo de Fin de Grado
<i>IMDEA</i>	Institutos Madrileños de Estudios Avanzados
<i>LED</i>	Light-Emitting Diode
<i>OLED</i>	Organic Light-Emitting Diode
<i>LEC</i>	Light-Emitting Electrochemical Cell
<i>DC</i>	Corriente continua
<i>AC</i>	Corriente alterna
<i>IDE</i>	Entorno de Desarrollo Integrado

PART I

MEMORIA



Chapter 1

Introducción

LA bombilla incandescente es una de las mejores invenciones del ser humano. Sin embargo, está siendo sustituida por tecnologías como lámparas fluorescentes, LEDs (*Light-Emitting Diode*) o LECs, *Light-Emitting Electrochemical Cell*. Un LEC es un material multicapa cuya capa activa tiene un material semiconductor que emite luz visible cuando se le aplica un voltaje [1, 2]. Estos últimos aún están en desarrollo. Parte del estudio de estos cristales LEC consiste en analizar el comportamiento de su impedancia para caracterizar eléctricamente y ópticamente a estos LEC. Esto se realiza inyectando una señal pulsada a diferentes frecuencias y monitorizando la corriente que pasa por el LEC. Para analizar la impedancia hay que diseñar un sistema que genere una señal pulsada a diferentes frecuencias, preferiblemente una señal sinusoidal. También se necesita que el sistema sea capaz de suministrar corrientes de 0mA a 100mA de corriente alterna. Este trabajo de fin de grado se centrará en el diseño del sistema y conectar un medidor de impedancia con los cristales LEC emitiendo un régimen pulsado. En la zona en la que el pulso de luz esté en OFF, se medirá la impedancia. Sin embargo no ha dado tiempo a conectar ambos sistemas y el trabajo se ha centrado en generar corrientes de 0 mA a 100 mA para frecuencias de 0 a 120 kHz para poder conseguir que los cristales LEC emitan luz. Una tarea futura será realizar la conexión entre sistemas y poder medir y detectar barridos de frecuencias de hasta MHz.

1.1. Motivación

Parte de la investigación de los cristales LEC consiste en el estudio del comportamiento de la impedancia de estos dependiendo de la frecuencia de la señal que se inyecte a los cristales. Este trabajo de fin de grado consiste en realizar un sistema que genere la señal pulsada y recoja los datos de la señal una vez pasa por el cristal LEC. Se hará un barrido en frecuencias, pero habrá un enfoque especial en frecuencias en torno a kHz. Es importante que la señal que se genere en el sistema sea capaz de suministrar corrientes de 0mA a 100 mA de corriente alterna. Se hará un posterior estudio del comportamiento del LEC a diferentes frecuencias de la señal inyectada al mismo y a diferentes valores de corriente. Se monitorizará la señal de entrada al LEC, el voltaje y la corriente que se suministra al cristal.

El sistema Botest es un sistema que mide impedancia a altas frecuencias. Se lleva midiendo desde hace tiempo la impedancia de estos cristales. Este proyecto tiene el objetivo de realizar un sistema a medida y abierto en función de las aplicaciones deseadas en IMDEA Materiales. En la actualidad ya se tiene un sistema hecho a medida que controla la potencia suministrada al LEC, monitorizando la corriente que pasa a través de él junto con su caída de voltaje. Esto se realiza con una tarjeta de adquisición de *National Instruments (USB6008)* que genera y recibe la señal pulsada o continua. Sin embargo, la corriente y voltaje que fluyen por el cristal no son mayores de 5 mA y de 5 V. Además, el barrido de frecuencias tiene un límite de 20kHz. El

principal problema con la tarjeta *USB6008* es que no se puede realizar medidas de impedancia a altas frecuencias.

El sistema que se diseñe en este trabajo de fin de grado servirá para sustituir al sistema diseñado con la tarjeta de *National Instruments*, mejorando así la potencia que se inyecta a la señal, aumentando considerablemente los valores de voltaje y corriente que fluyan por el LEC en cada momento. También se mejorará el barrido de frecuencias ya que el sistema debe ser capaz de generar señales con una frecuencia de hasta 120 kHz. Inicialmente el límite superior de frecuencia era 1MHz, pero no se consiguió llegar a éste límite y se ha rebajado la frecuencia superior hasta 120kHz. El motivo para rebajar esta frecuencia hasta 120 kHz es porque con los componentes utilizados no tenían un funcionamiento óptimo a frecuencias muy elevadas, como se explicará más adelante.

Los cristales LEC aún están bajo desarrollo e investigación y se quiere estudiar la impedancia de estos para conocer mejor el comportamiento iónico y electrónico del mismo. La medición de impedancia de un cristal LEC se realiza inyectando una señal pulsada (puede ser una señal otra señal, lo importante es conocer su fase) al cristal y después comparando las señales de corriente y voltaje y estudiando la fase que exista entre ambas. La impedancia se medirá a diferentes frecuencias de la señal pulsada, haciendo un barrido desde los 5 Hz hasta frecuencias del orden de 120 kHz, así que se busca desarrollar un sistema que se comuniquen con la placa que mide la impedancia de la señal y que controle la potencia de emisión del pulso.

Con este proyecto se busca realizar un prototipo simple, útil y fácil de manejar para la generación de pulsos y suministro de corriente de frecuencia a los LECs. La potencia que se inyecta a los LECs vendrá determinada por la corriente, de 5 mA a 100 mA y por el voltaje, que será de 0 V a 25V. Por tanto la potencia del sistema variará entre los 25 mW y los 3 W. Esta frecuencia se querría que pudiese alcanzar los MHz pero solo se ha conseguido que llegue a emitir a cientos de kHz. Una vez la señal haya fluido por el LEC, en un futuro desarrollo se medirá la impedancia del mismo con otro sistema. El control de potencia de emisión del LEC podrá ser monitorizado y la selección de frecuencias de emisión, así como la corriente suministrada en cada momento, podrá ser controlada desde una interfaz. De esta manera se podrá medir paralelamente la impedancia de los cristales en un estado de emisión bajo de los LECs sin tener que superponer señales: la producida para medir la impedancia y la necesaria para la emisión del LEC.

Un objetivo en el futuro es poder comunicar el comienzo de la emisión del LEC con la placa de adquisición que mida la impedancia del cristal cuando éste esté apagado. De esta forma, cuando el LEC esté en estado apagado durante el tren de pulsos de emisión, el sistema de medida de impedancia recibirá una señal de trigger para que comience a medir la impedancia del cristal en su estado de emisión OFF.

1.2. Objetivos

El proyecto, después de haber evaluado las dificultades del mismo y el tiempo restante, ha de cumplir los siguientes objetivos:

- Que el sistema sea capaz de generar una señal cuya frecuencia varíe entre los 5 Hz y 120 kHz, y cuya amplitud varíe de 0 V a 25 V. Es decir, que el sistema genere una señal sinusoidal con voltaje pico-pico de 0 V a 20 V. Este objetivo se puede desglosar en dos fases:
 - Generación de la señal con el chip AD9833. Este chip permite generar una señal sinusoidal cuya amplitud es de 330-350 mV y su frecuencia puede ser programada para variar entre 0 Hz y 12.5 MHz.

- Diseño e implementación de un circuito de amplificación cuya ganancia sea la suficiente para cumplir con las directivas de la señal que tiene que ser inyectada a los cristales LEC. Este circuito de amplificación también debe tener una respuesta en frecuencia válida a frecuencias de kHz, es decir, debe amplificar la señal correctamente en el rango de frecuencias exigido.
- Que el sistema sea capaz de recoger la señal una vez ha sido inyectada en el cristal LEC con el uso de un sistema de adquisición.
- Crear una interfaz amigable que permita controlar la potencia de emisión de LEC (corriente inyectada) y la frecuencia de emisión de la luz.

1.3. Recursos

Los recursos que se han empleado para el diseño final varían bastante de lo que inicialmente se escribió en el Anexo B. No obstante los nuevos recursos facilitan el desarrollo del sistema enormemente. A continuación se listarán los recursos que se han empleado para el desarrollo de este sistema.

Se puede observar que la gran mayoría de componentes que se han utilizado tienen pines de inserción. La elección de los mismos se debe a que uno de los objetivos del trabajo es realizar un entregable, cumpliendo así la fase de integración del sistema. Este entregable, entre otras cosas, contiene un circuito soldado en una placa. De ahí la elección de componentes con pines de inserción. Además, el circuito se ha probado en protoboards antes de soldarlo. De las pruebas se reciclan varios componentes para la soldadura posterior.

■ Hardware

- Arduino NANO. Microcontrolador que se usará para controlar diferentes dispositivos y componentes en el circuito. El Arduino NANO es de una de las tarjetas más pequeñas de Arduino. Se ha elegido un Arduino NANO en lugar de un Arduino UNO porque el NANO tiene prácticamente la misma funcionalidad que el UNO y además tiene pines de inserción, lo cual es una ventaja para el sistema desarrollado, pues parte del circuito quedará soldado en una placa. Además, un Arduino UNO estaría sobredimensionado para este proyecto.[3]

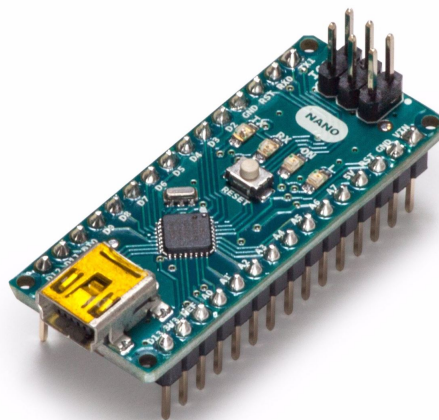


Figure 1. Arduino NANO

- Generador de señal AD9833. Dispositivo programable capaz de producir señales sinusoidales, triangulares y cuadradas a frecuencias de 0 Hz a 12.5 MHz. Es la tarjeta que se usa para generar la señal pulsada de entrada al sistema. [4]

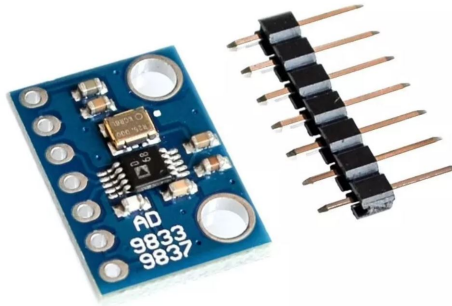


Figure 2. AD9833

- Multímetro Agilent. Este multímetro es la herramienta que se utiliza en este sistema para monitorizar la corriente que pasa por el LEC en todo momento. También se ha usado durante toda la fase de diseño y desarrollo del circuito para comprobar que las especificaciones de corriente se cumplieran. A pesar de que es un dispositivo caro es una opción muy cómoda para la monitorización del LEC. Tiene una resolución magnífica de 6,5 dígitos y su escala es muy buena: de nA a 2 A con una resolución de nA.[5]

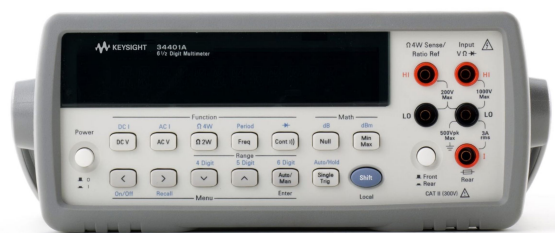


Figure 3. Multímetro Agilent

- Potenciómetro digital MCP41100. Se trata de un potenciómetro digital de 256 posiciones controlado por SPI. La resistencia nominal de este potenciómetro es de 100 k Ω . Un comando incluye el valor de la resistencia deseada en una palabra de 8 bits y se envía por SPI. Una vez el comando se ha procesado, el cursor del potenciómetro se mueve a la posición correcta para establecer la resistencia que pide el usuario. Sin embargo este potenciómetro no es un componente que admita altas potencias y esto ha causado fallos en las medidas del circuito.[6]



Figure 4. Potenciómetro MicroChip MCP41100 100 k Ω

- Amplificador operacional de potencia OPA445AP. La principal característica de este amplificador operacional es que su tensión de alimentación puede aumentar hasta los 45V, aunque para este sistema estos amplificadores se han alimentado a $\pm 25V$. También es importante anotar que este amplificador operacional tiene un ancho de banda de 1 MHz, lo que significa que funcionará perfectamente en el barrido de frecuencias que exige este trabajo de fin de grado. Debido a que es un amplificador de alta potencia y tiene un ancho de banda considerable, es un componente caro, en torno a 10 euros la unidad. [7]

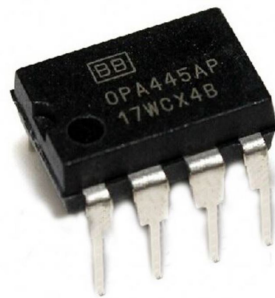


Figure 5. Amplificador operacional OPA445AP

- Amplificador operacional TL072A. Se trata del amplificador operacional que se utilizó en las primeras iteraciones del proyecto. Sin embargo, la tensión de alimentación de este operacional era demasiado baja para los requisitos de potencia que el circuito pedía.
- Transistor MJE3055T. Transistor NPN de alta potencia. Este transistor se utiliza en la última etapa del circuito y juega un papel muy importante en el funcionamiento del sistema. Su máxima corriente de colector I_c es de 10 A y el ancho de banda máximo es de 2 MHz. Tiene un voltaje máximo entre el colector y el emisor de 60 V y disipa una potencia máxima de 75 W en el colector. Más adelante se explicará la elección de este transistor. [8]

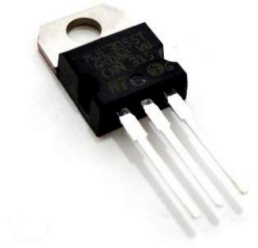


Figure 6. Transistor MJE305

- Switch analógico ADG444. Contiene 4 interruptores que se activarán o desactivarán según la instrucción que obtengan del Arduino NANO. [9]

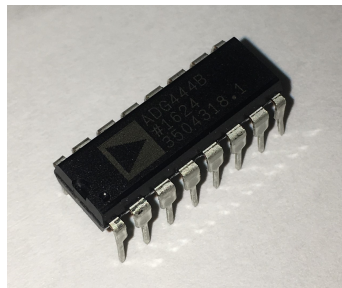


Figure 7. Switch analógico ADG444

- Potenciómetro analógico de potencia. Este potenciómetro tiene una resistencia nominal de $500\ \Omega$ y una potencia de $40\ W$. Se ha utilizado durante el diseño del circuito para elegir qué valor de resistencia se pondría en el emisor del transistor MJE3055T. Como la última etapa del circuito genera unos voltajes moderadamente elevados es necesario utilizar una resistencia de potencia.



Figure 8. Potenciómetro de potencia



Figure 9. Potenciómetro de potencia

- Resistencia de potencia ARCOL de 250 Ω . Resistencia de potencia que se adquirió después de usar el potenciómetro de potencia. La potencia máxima que puede disipar esta resistencia es de 50W. [10]

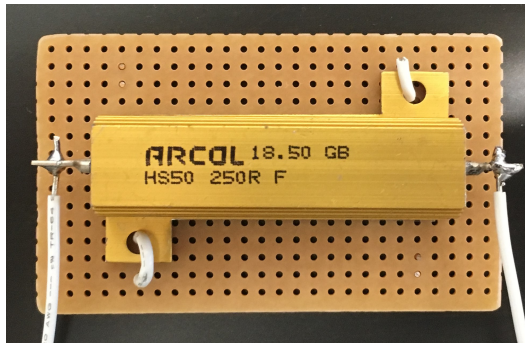


Figure 10. Resistencia de potencia

- Resistencias de 1/4 W. Resistencias convencionales que se utilizan para las etapas de filtrado y amplificación. En concreto, se utilizan los siguientes valores nominales de resistencia:
 - 1k Ω .
 - 10k Ω .
 - 12k Ω .
 - 27k Ω .
 - 1M Ω .



Figure 11. Resistencias

- Condensadores. Se utilizan para etapas de filtrado. Los condensadores que se han utilizado son:
 - Condensador electrolítico de $0.1\mu\text{F}$.
 - Condensador de 100pF .



Figure 12. Condensador electrolítico de $0.1\mu\text{F}$



Figure 13. Condensador de 100pF

- Pines header macho. Se utilizan para la soldadura del circuito, ya que se ha decidido conectar los elementos del circuito con cables dupont hembra-hembra en lugar de soldar cables y hacer el circuito difícil de *debuggear*.



Figure 14. Pines header macho

- Cables dupont hembra-hembra. Para conectar elementos del circuito conectándolos a los pines header macho.

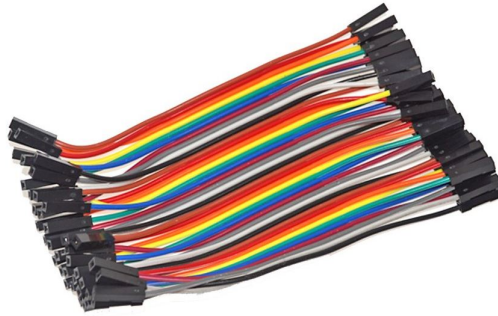


Figure 15. Cables dupont hembra-hembra

- Clemas para soldar la alimentación a una placa de soldadura experimental.



Figure 16. Clema

- Placa para soldar. Placa de baquelita experimental para soldar el circuito una vez se haya comprobado que funciona en una protoboard.

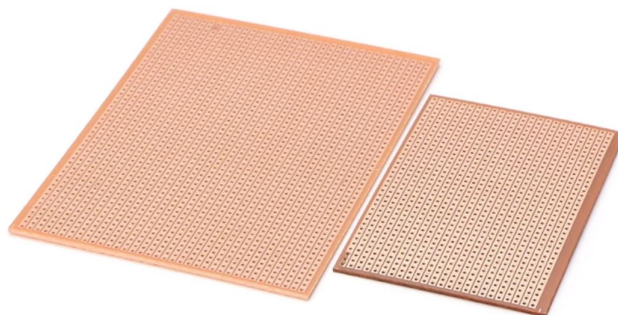


Figure 17. Placa para soldar el circuito

- Soldador y estaño. Para soldar los componentes del circuito a la placa.



Figure 18. Soldador y estaño

- **Software.** El software que se ha utilizado para este proyecto es esencialmente para el desarrollo de programas para controlar el circuito.
 - Software de Arduino. Es el IDE oficial de Arduino donde se han programado los scripts necesarios para que los dos Arduinos NANO que se utilizan en el sistema realicen las tareas debidas.
 - Software de MATLAB. Entorno de desarrollo de MathWorks para computaciones matemáticas. Se ha utilizado para escribir funciones de cálculo de resistencias en función de qué corriente desee el usuario y generación de comandos a enviar a los Arduinos. También se ha utilizado el *AppDesigner* que viene incluido en MATLAB para diseñar la interfaz gráfica con la que va a interactuar el usuario y darle la funcionalidad necesaria.
 - LabVIEW Software. Se trata de un software orientado a la ingeniería e investigación que permite diseñar aplicaciones para hacer pruebas, medidas y control muy poderosas. Este software no es de código abierto así que hay que comprar una licencia para poder usarlo. Para este trabajo se ha utilizado la licencia que IMDEA Materiales tenía comprada y utiliza para otros proyectos.

1.4. Metodología

Los objetivos descritos en el apartado anterior se conseguirán siguiendo una metodología que incluirá fases de investigación, implementación del prototipo y evaluación del prototipo. Estas últimas dos fases se repetirán tantas veces como sean necesarias hasta conseguir un modelo que cumpla adecuadamente los objetivos impuestos y responda a la motivación del proyecto. El ciclo final del desarrollo del modelo incluirá un prototipo entregable que se pueda usar de inmediato.

En cuanto a la parte de investigación, ha sido extensa y abarca cómo configurar los diferentes recursos que se emplean. Parte de la investigación incluye la programación de las diferentes placas a utilizar, como el chip AD9833 o los sketches de Arduino y el análisis de la funcionalidad de cada una de ellas.

En el paso previo al desarrollo se ha realizado un diagrama de bloques del sistema en su totalidad para identificar y definir claramente las funcionalidades de cada bloque y aislar los bloques entre ellos, para asegurar su funcionamiento independientemente de los bloques anteriores o posteriores. Es decir, que cada bloque realice su funcionalidad adecuadamente.

Una vez se ha diseñado un diagrama de bloques adecuado, se define en más detalle cada uno de los bloques. Por ejemplo, qué alimentación deberá recibir cada placa por separado o el diseño del circuito amplificador, con su respuesta en frecuencia y su ganancia correctamente identificadas y definidas.

El desarrollo del sistema, siguiendo el paso anterior, se ha hecho por bloques. Aunque no es necesario seguir un orden concreto a la hora de desarrollar los distintos bloques que componen el sistema, sí que es una buena idea ir desarrollando cada bloque de manera secuencial, para una vez desarrollado cada bloque, poder acoplarlo al anterior (o a los bloques anteriores) y evaluar su funcionamiento conjunto.

Por último, se acoplarán todos los bloques para componer el entregable que exige el proyecto y se analizarán los resultados obtenidos al utilizar el sistema.

La metodología del diseño, desarrollo e implementación del sistema se puede desglosar en varias actividades:

- Actividad 1. Investigación extensa acerca de los recursos a usar en el proyecto. Inicialmente se habían seleccionado las placas EVAL-SDP-CB1Z, EVAL-ADUC7020QSZ y EVAL-AD9833SDZ para la generación de la señal y la adquisición de la misma una vez hubiese fluido por el cristal LEC. Sin embargo, durante el diseño del circuito estas placas presentaron varios problemas de conectividad con el ordenador que las controlaba y se descartaron, dando paso al uso del chip AD9833, que es el mismo chip que contiene la placa EVAL-AD9833SDZ sin necesidad de utilizar una tarjeta programadora como es la placa EVAL-SDP-CB1Z.
- Actividad 2. Familiarización y configuración de los recursos a utilizar. Diseño de un programa en Arduino que controle el chip AD9833 para generar una señal sinusoidal con la frecuencia que se precise en cada momento.
- Actividad 3. Diseño del diagrama de bloques. Al ver que el proyecto iba a requerir más tiempo del que se tenía se pasó de medir MHz a medir cientos de kHz pues los amplificadores no funcionaban bien a frecuencias más altas. También por falta de tiempo se decidió centrarse en controlar solamente la corriente. No se trabajó tampoco con el medidor de impedancia ni se midió la zona baja del pulso. Así que el trabajo se convirtió en realizar un sistema de alta potencia con las dificultades que esto conlleva.
- Actividad 4. Implementación del diseño.
 - Actividad 4.1. Generación de la señal sinusoidal.
 - Actividad 4.2. Diseño y desarrollo del circuito amplificador para conseguir la potencia necesaria de emisión de LEC. Esta actividad ha llevado mucho tiempo, ya que se han implementado diferentes diseños que no cumplían con los requisitos pedidos del sistema y ha habido que descartar muchos circuitos. Por ello el proyecto se ha limitado a construir una fuente de corriente alterna.
 - Actividad 4.3. Diseño y desarrollo de la interfaz gráfica para seleccionar qué señal se quiere inyectar en cada momento y qué corriente queremos que el sistema suministre al cristal LEC.
- Actividad 5. Integración del sistema. Se soldará el circuito a una placa experimental y se introducirá en una caja portátil y de reducido tamaño.
- Actividad 6. Análisis de resultados y evaluación del prototipo.
- Actividad 7. Redacción de la memoria del proyecto.

1.5. Organización del documento

El documento se ha organizado por bloques, como por ejemplo, montaje del circuito, resultados, conclusiones, manual de usuario, etc.

El documento comienza con el **estado del arte**, donde se explica qué es un cristal LEC, para qué se está utilizando ahora mismo y por qué interesa construir un sistema que se pueda aplicar al cristal LEC. Después, se procede a **describir el sistema** en cuestión. Primero se discute la generación de la señal pulsada, después el diseño del circuito y cada una de sus etapas, la interfaz del usuario y finalmente el funcionamiento del conjunto completo.

Una vez se ha descrito el sistema se procede a explicar su **montaje** y cómo se ha integrado en una caja fácil de utilizar. El usuario interactuará con el sistema a través de las entradas y salidas que presenta la caja y mediante la interfaz de usuario diseñada.

Los **resultados** del trabajo vienen a continuación. Se describen los resultados y se enumeran las dificultades y problemas encontrados durante el diseño y montaje del sistema. También se proponen soluciones a varios de los problemas enunciados, así como **futuros desarrollos y mejoras**.

Finalmente se ha escrito un pequeño **manual de usuario** para aquella persona que quiera utilizar el sistema, se cerciore de conectar las entradas adecuadamente y colocar bien la polaridad del LEC al acoplarlo al sistema.

Se adjunta también un apartado con los **extractos de código** que se han escrito para el trabajo.

Por último, al final de la memoria hay un apartado que incluye todo el código que se ha escrito para este proyecto, bajo el nombre de **código fuente**.

Chapter 2

Estado del Arte

EN este capítulo se procederá a explicar el **estado de la técnica**, es decir, qué es un cristal LEC, para qué sirve, cómo está hecho y cómo está el campo de conocimiento del LEC en el momento en el que se realiza este trabajo de fin de grado.

2.1. Introducción

Como mencionado en capítulos anteriores, la bombilla incandescente ha sido uno de los inventos más revolucionarios del ser humano y ha monopolizado el mercado de la iluminación durante décadas. Sin embargo, está siendo destronada por tecnologías más avanzadas como lámparas fluorescentes o LEDs (*light-emitting diode*). El mayor tiempo de vida y la mejor eficiencia de estas nuevas fuentes de iluminación ha llevado a la caída de la bombilla tradicional. Aún así, hay que puntualizar que estas nuevas fuentes de iluminación emergentes son buenas, pero no perfectas. La electrónica detrás de su fabricación es muy compleja y los procesos de fabricación son clasificados como de riesgo y dañinos para la salud, debido al alto contenido de metal que estos dispositivos contienen. Todo esto implica que hacer el producto final es un proceso costoso.

Aunque tradicionalmente no han sido clasificados como electrónicamente activos, algunos materiales orgánicos se pueden comportar como semiconductores inorgánicos. La conductividad de estos materiales se consigue a través del llamado **dopaje**. Por ejemplo, hay ciertos polímeros que pueden adquirir propiedades metálicas si se les aplica el dopaje.

Además de las lámparas fluorescentes y los LED, existen dispositivos como los OLED o LCD. El OLED (*Organic Light-Emitting Diode*) compite con el LCD (*Liquid Crystal Display*). Los OLED tienen un proceso de fabricación muy simple y barato que hace de ellos un dispositivo muy competitivo en el mercado. Además, ofrecen un rendimiento muy bueno de display, consiguiendo colores más vívidos y con buenos contrastes. Los OLED tienen la capacidad de competir con los LCD porque no necesitan luz de fondo o *backlight* para funcionar y por ello consumen menos potencia. Sin embargo, **los OLED de alto rendimiento** necesitan películas muy finas, estructuras multicapa y constitutivos sensibles al aire: estructura y métodos de fabricación complejos que alejan al OLED de alto rendimiento de ser la opción más barata del mercado.

Los cristales LEC (*Light-Emitting electrochemical Cell*) son una alternativa más eficiente y de bajo coste a los cristales OLED. Fueron desarrollados por Pei, Q et al. en 1995. [1]

2.2. Qué es un LEC

Un LEC es un material con varias capas: una capa de sustrato que puede ser aluminio o cristal, dos electrodos, y la capa activa, contenida entre los electrodos. La capa activa consiste en un material semiconductor electroluminiscente, orgánico o inorgánico, que emite luz visible cuando se le aplica un campo eléctrico.



Figure 19. LEC que emite un mensaje

Los cristales LEC funcionan gracias a la presencia de iones móviles en su capa electroluminiscente que promueve el dopaje electroquímico del emisor a niveles bajos de voltaje.

Comparados con los OLEDs, los cristales LEC (*Light-Emitting Electrochemical Cell*) ofrecen mayor resistencia contra superficies mas bastas y rugosas y defectos de las películas, permitiendo fabricar dispositivos de mayor área de manera más sencilla y con métodos escalables. Un LEC puede fabricarse usando materiales estables en aire y no depende de estructuras multicapa, lo cual simplifica su fabricación sustancialmente. Por esto, los cristales LEC pueden tener una probabilidad mayor de alcanzar las expectativas de que la electrónica orgánica pueda crear dispositivos prácticos e imprimibles a muy bajo coste.[1, 2, 11, 12, 13]

2.3. Turn-on time y lifetime del LEC

2.3.1. Lifetime o tiempo de vida del LEC

El tiempo de vida de los cristales LEC es un buen indicador al evaluar la durabilidad del mismo para aplicaciones de iluminación o display. El tiempo de vida de un LEC se define como el **tiempo que tarda la luminosidad del dispositivo en decaer desde el máximo de luminosidad hasta la mitad del máximo de luminosidad**.

2.3.2. Tiempo de encendido o device turn-on time

Al aplicar al cristal LEC un **voltaje de polarización** o *bias voltage*, la densidad de corriente y la luminosidad del LEC aumentan con el tiempo debido al movimiento y redistribución de los iones libres dentro del cristal LEC.

El tiempo de encendido de un cristal LEC, (*turn-on time*) se define como el tiempo que tarda el LEC en alcanzar la máxima luminosidad mientras se le aplica un voltaje de polarización constante. El tiempo de encendido puede estar entre unos pocos minutos y varias horas. Este tiempo se considera un tiempo bastante largo y para aplicaciones prácticas sería interesante

acortarlo. El único problema es que acortar el tiempo de encendido se suele conseguir a expensas de la estabilidad del dispositivo, haciéndolo así mucho más inestable.

Aplicar voltajes de polarización mayores, ya sean en corriente alterna o corriente continua implica un tiempo de encendido del dispositivo menor, ya que voltajes mayores causan un movimiento de los iones hacia los electrodos mucho más rápido.

Se ha comprobado [12] que aplicar un voltaje de polarización en forma de señal pulsada mejora la estabilidad del dispositivo y el tiempo de encendido del mismo.

El tiempo de encendido también depende del grosor de la capa de emisión de luz del cristal LEC. Capas más finas tendrán un tiempo de encendido menor mientras que capas más gruesas tendrán un tiempo de encendido mayor. También se ha observado que las condiciones atmosféricas afectan muchísimo al funcionamiento del dispositivo, tiempo de encendido, nivel de luminosidad, etc. Los cristales LEC desarrollados en IMDEA Materiales tienen mejor rendimiento en cámaras de vacío que al aire. [12, 13]

2.4. Formación de la unión p-n

Cuando se aplica un voltaje a los cristales LEC por medio de los electrodos, los iones móviles generan regiones p- y n- dopadas conectadas por una región no dopada. Estas regiones comienzan a crecer y la zona no dopada se va reduciendo. Después de un cierto tiempo de haber aplicado el voltaje al cristal LEC, las zonas p- y n- se unen y la zona no dopada desaparece, por lo que el cristal LEC deja de emitir luz. [1, 2]

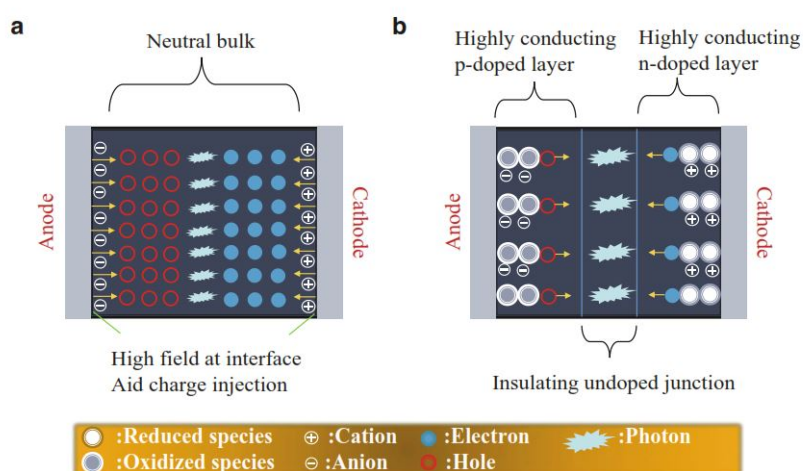


Fig. 4.1 Schematic diagrams for operation mechanisms of (a) electrodynamical (ED) model and (b) electrochemical doping (ECD) model

Figure 20. Formación del dopaje p-n

2.5. Arquitectura del dispositivo LEC

Un cristal LEC contiene dos electrodos y una capa activa con iones móviles y un semiconductor orgánico que emite luz. El LEC puede tener varias estructuras, pero las más comunes son la **célula de superficie** o *surface cell* y la **célula en sándwich** o *sandwich cell*. La célula en sándwich es la más práctica ya que tiene una superficie de emisión de luz mayor, delimitada por la superposición de los electrodos superior e inferior. En la célula de superficie,

la zona de emisión de luz tiene lugar entre los dos electrodos que se encuentran en los extremos del cristal. [12]

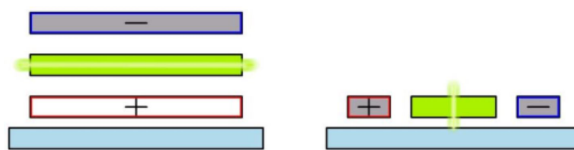


Figure 21. Célula sándwich y célula de superficie

2.6. Materiales de los electrodos

La conductividad de los electrodos en un cristal LEC es evidentemente importante, ya que una resistencia alta en los electrodos puede llevar a caídas de voltaje a lo largo de la superficie del electrodo durante su funcionamiento, que desemboca en una menor eficiencia y emisión de luz no uniforme. Los electrodos que utiliza IMDEA Materiales son de pintura de plata para asegurar su conductividad. [12]

2.7. El cristal LEC de IMDEA Materiales

En IMDEA Materiales están desarrollando sus propios cristales LEC. Para realizar pruebas con el circuito que se ha diseñado para este TFG algunos de los cristales de desarrollos anteriores fueron utilizados. Se adjuntan imágenes de cristales LEC que se han estado usando:

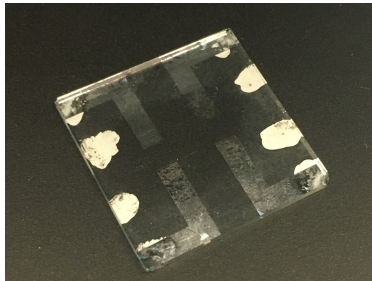


Figure 22. LEC desarrollado en IMDEA

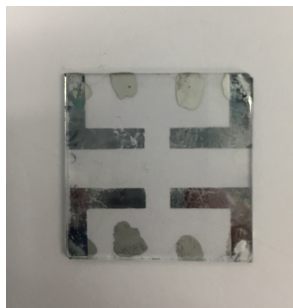


Figure 23. LEC desarrollado en IMDEA

El cristal que se muestra en las imágenes superiores está formado por 4 LECs. Cada uno de los LECs se encuentra en el extremo interior de la película metálica que tiene forma de "L". Las manchas blancas que se observan en las aristas del cristal corresponden con pintura de plata.

La pintura de plata funciona a modo de electrodo. Así se podrá inyectar la señal que genera el sistema.

2.8. Característica V-R del LEC

Como se ha mencionado previamente, en IMDEA Materiales están desarrollando sus propios cristales LEC. Durante la investigación se han tomado resultados de las primeras iteraciones del diseño. Se ha proporcionado la siguiente tabla, que relaciona la resistencia del cristal LEC con el voltaje que se aplique entre sus electrodos. Como se puede observar en la figura, a 0 V el LEC tiene una resistencia de 100 k Ω , disminuye hasta 10 k Ω a 3 V y finalmente a 5 V tiene una resistencia de 300 Ω , aproximadamente. Estas medidas ayudan a entender el cristal LEC como el material semiconductor que va a actuar como carga no lineal del circuito que se ha diseñado.

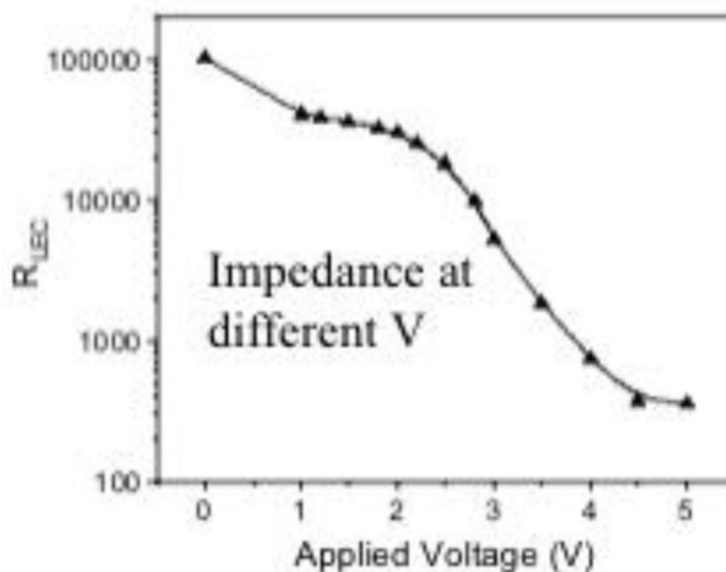


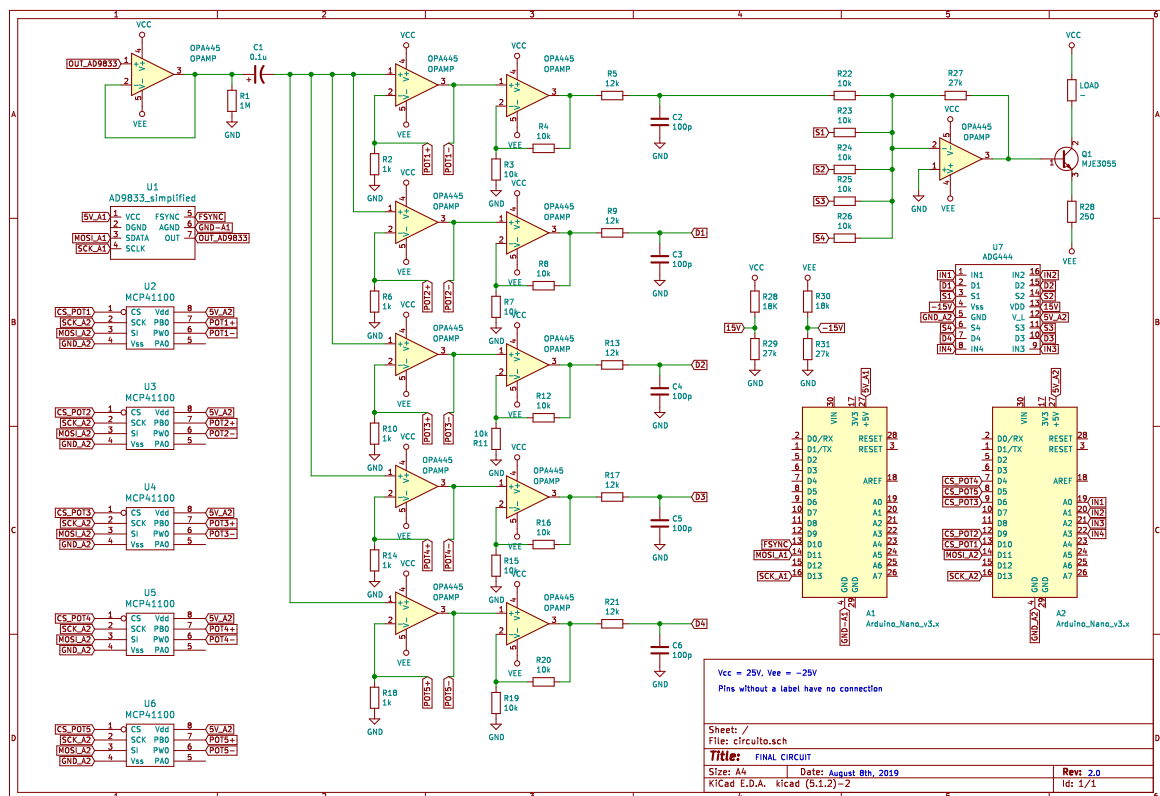
Figure 24. Característica V-R del LEC

Se ha tomado como principal punto de referencia la resistencia del LEC a 5V, para así poderla sustituir en los cálculos por una resistencia fija cuyo valor nominal sean 300 Ω . Se han utilizado estos datos de resistencia de un desarrollo anterior de los cristales y no se conoce el valor de la resistencia de los LEC que están ahora mismo en desarrollo. Se espera que estas medidas sean un buen punto de referencia para el desarrollo del sistema y para que éste se pueda utilizar para medir el comportamiento de la impedancia de los cristales LEC en desarrollo.

Chapter 3

Descripción del Sistema

A continuación se adjunta el plano esquemático del circuito al completo. Consta de una etapa de generación de señal, varias etapas de filtrado y amplificación, conexión con Arduinos, etc. Las etapas se discutirán más en detalle en este capítulo.



3.1. Generación de la señal con el AD9833

El AD9833 es un generador de ondas programable de baja potencia. El AD9833 es un chip DDS (*direct digital synthesis*) integrado que requiere de un reloj, una resistencia de baja precisión y condensadores de desacoplo para poder crear señales sinusoidales de 0 Hz hasta 12.5 MHz. [4]

Es un dispositivo capaz de producir señales sinusoidales, triangulares y cuadradas. Además de poder programar el tipo de onda, también es posible programar la frecuencia y la fase de la misma.

La fase y la frecuencia se pueden escribir en los registros `FREQ0`, `FREQ1`, `PHASE0`, `PHASE1`.

El AD9833 consiste de las siguientes partes:

- *Numerically controlled oscillator (NCO).*
- Moduladores de fase y frecuencia.
- SIN ROM.
- DAQ (conversor analógico digital).
- Regulador de tensión.

El AD9833 funciona vía interfaz serial. El dispositivo hace uso de un reloj externo para escribir datos o información de control al dispositivo. Este reloj serial externo puede tener una frecuencia de hasta 40 MHz. En este caso se ha programado un Arduino NANO y escribiendo un programa en Arduino para enviar los datos por la interfaz Serial del Arduino usando SPI. La librería que se ha usado para darle la funcionalidad de AD9833 al programa se ha tomado de GitHub. [17]

El dispositivo funciona gracias a las 3 entradas serial, `SCLK`, `FSYNC` y `SDATA`. `SCLK` (*serial clock input*) se usa como input de control para cargar los datos al dispositivo. La entrada `FSYNC` es una entrada *level-triggered* que funciona como *chip enable* y sincronización de *frames*. Los datos solamente pueden ser enviados cuando `FSYNC` está a nivel bajo. Los datos se cargan como una palabra de 16 bits por el input `SDATA`. Cuando `FSYNC` está a nivel bajo, los datos se cargan al *shift register* de entrada en los flancos de bajada de `SCLK` durante 16 pulsos (ya que es una palabra de 16 bits). Dependiendo de cómo lo programemos, `FSYNC` puede ponerse a nivel alto en cuanto termine de cargarse la palabra de 16 bits o puede mantenerse a nivel bajo durante un múltiplo de 16 `SCLK` pulsos y luego ponerlo a nivel alto al terminar la transferencia de datos. Esta última modalidad permite enviar un flujo continuo de palabras de 16 bits.

El chip se debe alimentar para que funcione. Se alimenta a 5V que es el voltaje que suministra la tarjeta de Arduino que va a controlarlo.

3.1.1. Imágenes

Las figuras que se muestran a continuación son capturas de la pantalla del osciloscopio, midiendo la salida de la tarjeta AD9833 siendo programada a diferentes frecuencias: 1kHz, 2.5kHz, 10kHz, 25kHz, 50kHz, 100kHz, 500kHz y 1MHz.

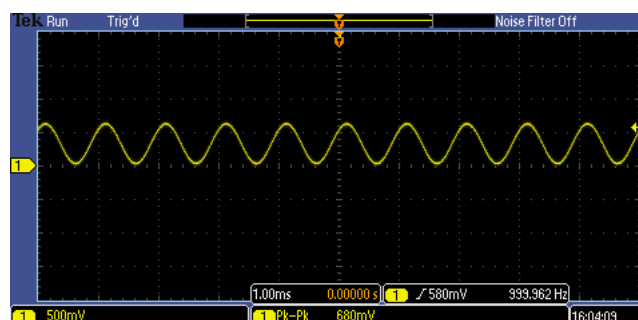


Figure 25. Señal de frecuencia 1 kHz generada por el AD9833

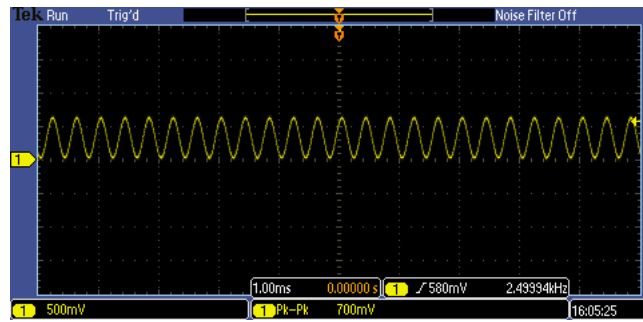


Figure 26. Señal de frecuencia 2.5 kHz generada por el AD9833

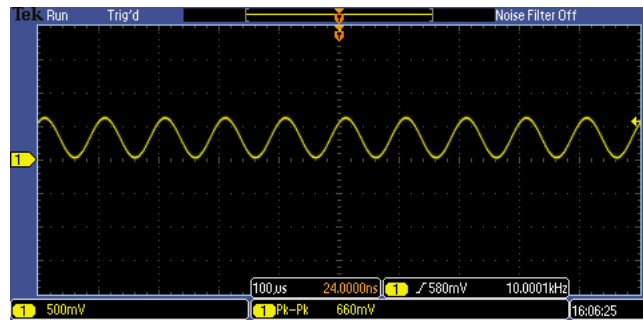


Figure 27. Señal de frecuencia 10 kHz generada por el AD9833

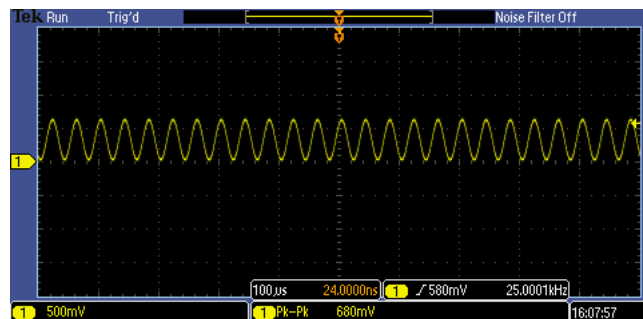


Figure 28. Señal de frecuencia 25 kHz generada por el AD9833

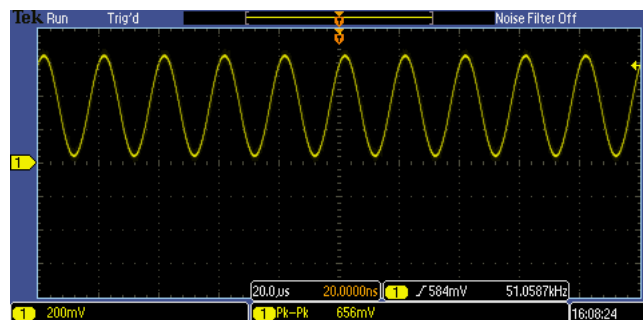


Figure 29. Señal de frecuencia 50 kHz generada por el AD9833

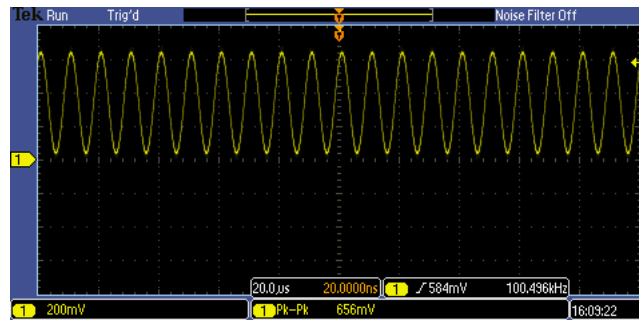


Figure 30. Señal de frecuencia 100 kHz generada por el AD9833

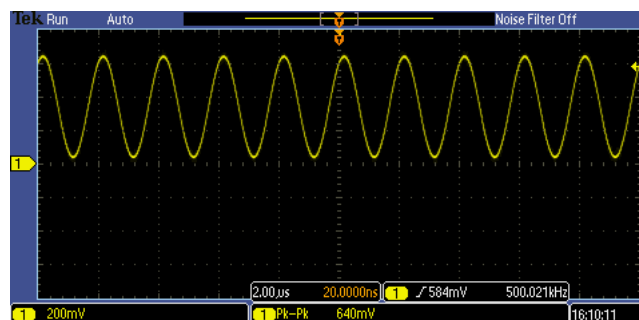


Figure 31. Señal de frecuencia 500 kHz generada por el AD9833

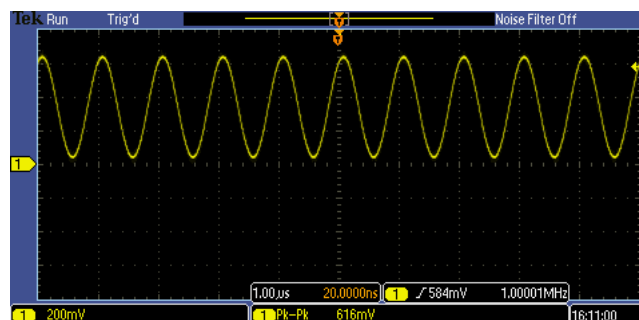


Figure 32. Señal de frecuencia 1 MHz generada por el AD9833

La señal de salida de que nos interesa para este TFG es la sinusoidal. Para la modalidad sinusoidal la señal que genera este chip es de unos 700 mV de voltaje pico pico. Esto significa que la señal de salida tiene un valor medio de alrededor de 330 - 350 mV y una amplitud de unos 350mV.

Se ha observado que a medida que aumenta la frecuencia de la señal disminuye la amplitud de la misma. Para barridos de frecuencia de hasta 120 kHz la disminución de amplitud es prácticamente inapreciable, pero cuando la señal generada es del orden de MHz hay una disminución considerable de la amplitud. A frecuencias mayores de 10 MHz la amplitud ha disminuido demasiado como para utilizarla en nuestro sistema.

3.2. El circuito

Los requisitos de este sistema es que el mismo suministre una corriente de 5 a 100 mA y una señal de voltaje de amplitud de 0 V a 25 V. También es necesario que el circuito funcione en un

rango de frecuencias de 5 Hz a 120 kHz. Para conseguir cumplir estos requisitos la señal que genera el AD9833 ha sido tratada por diferentes etapas de amplificación, filtrado y sumado y, por último, la señal se introduce en la base de un transistor para conseguir amplificar la corriente hasta los valores que se piden.

Como se ha visto en la sección anterior el AD9833 genera una señal sinusoidal de 0 a 700mV, más o menos. Eso quiere decir que hay un valor medio de la señal de 350 mV. Es preciso eliminar este valor medio antes de las etapas de amplificación ya que de lo contrario las señales a la salida de los amplificadores podrían estar saturadas debido a la presencia de un offset en la entrada del amplificador.

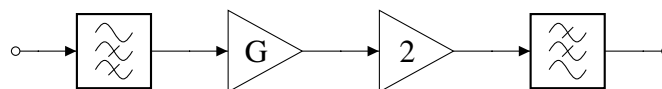
El circuito que se ha diseñado tiene componentes tanto analógicos como digitales, y en ciertas instancias del circuito se mezclan. Hay que recordar que la corriente que pase por el LEC tiene especial importancia. Como mencionado anteriormente, los requisitos del circuito son que suministre una corriente de 0 a 100 mA con la mayor resolución posible y que la tensión que pase por el LEC sea mayor a 20 V, aunque tiene mayor prioridad que el circuito se comporte como una fuente de corriente programable para el LEC y que dicha corriente pueda variar de 0 a 100 mA a petición del usuario del sistema.

Para analizar este circuito hay que dividirlo en varias etapas y comprender bien cada etapa antes de pasar a la siguiente. Sin contar con la generación de la señal con el chip AD9833 y considerando que ya tenemos una señal sinusoidal externa, el circuito puede dividirse en las siguientes fases: [14, 15]

- Filtrado paso alto.
- Etapa de amplificación de ganancia variable.
- Etapa de amplificación de ganancia fija.
- Filtrado paso bajo.
- Etapa final o transistor como fuente de corriente.

3.2.1. Diagrama de bloques del subcircuito: filtrado y amplificación

En el circuito hay un subcircuito que se repite cinco veces: las etapas de filtrado y amplificación son las que se repiten en cinco ocasiones diferentes. El diagrama de bloques del circuito que se repite es el siguiente:



Las etapas se procederán a describir en detalle a continuación.

3.2.2. Etapa 1: Filtrado

La primera etapa del circuito es un filtro paso alto. El AD9833 genera una señal de unos 600-700 mV de voltaje pico-pico y un offset de alrededor de 300 mV. Como las siguientes etapas van a ser de amplificación, se elimina la componente continua para trabajar solamente con corriente alterna. Esto evitará saturación en la señal de salida debido a la presencia de componente continua en la entrada. Si existe saturación en la señal de salida se podrá descartar que se deba a la tensión de offset en la entrada de la etapa amplificadora.

Uno de los requisitos del sistema es que haga un barrido de frecuencias de 0.1 Hz hasta 120 kHz. Que la frecuencia inicial del barrido sea 0.1 Hz también es un motivo por el cual eliminar el offset de la tensión de entrada.

Para eliminar el offset DC se construye el siguiente filtro paso alto:

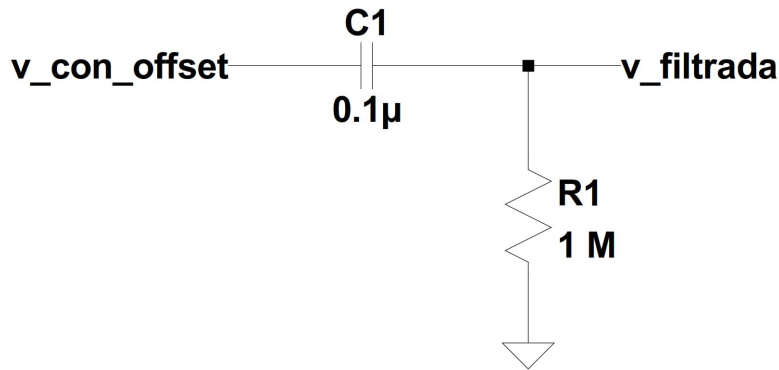


Figure 33. Red paso alto

La frecuencia de corte de este circuito se calcula con la siguiente fórmula:

$$f_c = \frac{1}{2\pi \cdot R \cdot C}$$

Los valores de resistencia y condensador de esta red paso alto son de $0.1 \mu F$ y de $1 M\Omega$. Con una frecuencia de corte tan baja se garantiza que únicamente se elimine el armónico en frecuencia 0, es decir el offset, sin alterar a las frecuencias cercanas como 1 Hz o 10 Hz.

$$f_c = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 1M\Omega \cdot 0.1\mu F} = 1.592Hz$$

Sin embargo, la señal de salida del sistema presenta nivel medio, a pesar de haberlo eliminado en esta etapa. El motivo por el cual la señal de salida presenta nivel medio se explica en la sección 5.2.2.

3.2.3. Etapa 2: Amplificación

Después del filtrado se procede a la amplificación de la señal. En la etapa de amplificación hay dos fases: primero, una etapa de amplificación con ganancia variable, controlada por el potenciómetro MCP41100, y una etapa de amplificación de ganancia fija 2 (V/V). Ambas etapas de amplificación son etapas no inversoras. Este último apunte no afecta al diseño del circuito o a la señal final de salida pero es importante anotarlo.

3.2.3.1. El potenciómetro digital MCP41100

El patillaje del potenciómetro es el siguiente:

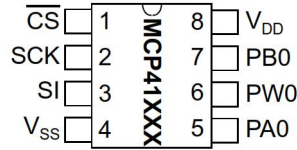


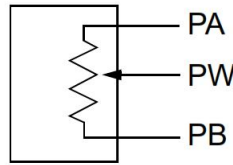
Figure 34. Patillaje del potenciómetro MCP41XXX

PB0, PW0 y PA0 son los tres terminales del potenciómetro. Entre PB0 y PA0 hay 100 kΩ de resistencia.

3.2.3.2. Funcionamiento de MCP41100

Las librerías de funcionamiento del potenciómetro MCP41100 en Arduino se han tomado de un repositorio público de GitHub. [18]

Para calcular la resistencia R_{WA} (resistencia entre el terminal A y el cursor) o R_{WB} (resistencia entre el terminal B y el cursor) se utilizan las siguientes fórmulas:



$$R_{WA}(D_n) = \frac{(R_{AB})(256 - D_n)}{256} + R_W$$

$$R_{WB}(D_n) = \frac{(R_{AB})(D_n)}{256} + R_W$$

Figure 35. Fórmulas para el cálculo de resistencias entre los terminales

Donde:

- R_{AB} es la resistencia total del potenciómetro. En nuestro caso, como mencionado anteriormente, 100 kΩ.
- R_W es la resistencia nominal del cursor, que según la hoja de datos es de 125 Ω.
- D_n es el valor de la resistencia deseada en una palabra de 8 bits.

En el sistema se han utilizado los potenciómetros tomando la resistencia R_{WB} en cada uno de los potenciómetros para emplearla como resistencia de ganancia de esta etapa amplificadora, como se explicará más adelante.

$$R_{WB}(D_n) = \frac{R_{AB} \cdot D_n}{256} + 125$$

Donde D_n es un dígito de 8 bits que se envía por SPI al potenciómetro. En este sistema se va a utilizar la interfaz serial del Arduino para mandar el valor de resistencia que deseamos

que tome el potenciómetro en cada caso. Por ello, en lugar de calcular la resistencia, habrá que calcular el dígito de 8 bits D_n a enviar al potenciómetro. El cálculo de D_n se realiza despejando D_n de la ecuación anterior obteniendo:

$$D_n = \frac{R_{desired} \cdot R_W}{R_{max,pot}} \cdot 256$$

Desgraciadamente, se ha comprobado que la presencia de un potenciómetro digital en la etapa de amplificación del circuito afecta a la señal que hay a la salida, causando un recorte a la misma. Este problema se explica en el capítulo 5 de esta memoria. No se tiene la certeza de que el recorte sufrido esté causado por el potenciómetro pero se considera una opción muy probable. Además, se desconoce si estos potenciómetros son de alta potencia y no se ha llegado a encontrar (por falta de tiempo principalmente) una alternativa a este potenciómetro digital pero de alta potencia.

3.2.3.3. Diagrama de la primera etapa amplificadora

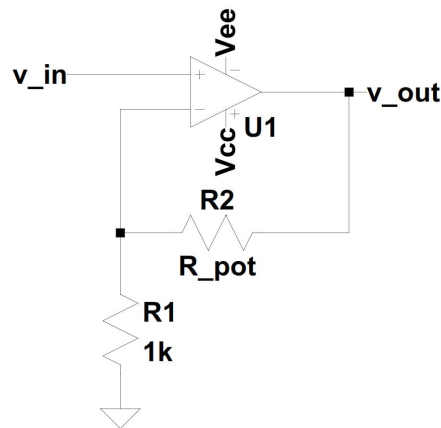


Figure 36. Etapa amplificadora nº 1

La fórmula para el cálculo de ganancia de esta etapa es la siguiente:

$$G = 1 + \frac{R_{WB}}{1k\Omega}$$

Siendo R_{WB} la resistencia que hay entre el cursor y el terminal B del potenciómetro.

3.2.4. Etapa 3: Amplificación

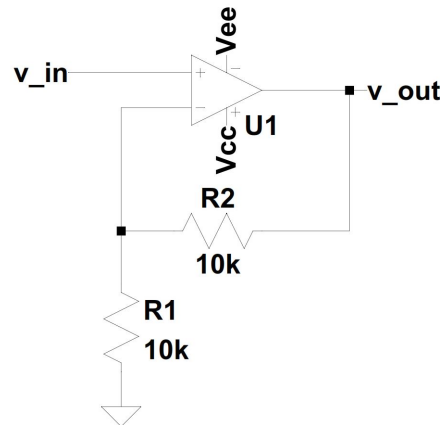


Figure 37. Etapa amplificadora nº 2

La segunda etapa de amplificación es una etapa de ganancia fija. Se consigue construyendo una etapa de amplificación no inversora cuyas resistencias son ambas de 10 kΩ. Con esto, la ganancia fija de esta etapa es:

$$G = 1 + \frac{10k\Omega}{10k\Omega} = 2\left(\frac{V}{V}\right)$$

Se añade una segunda etapa de amplificación al circuito porque con una única etapa de amplificación variable no se conseguía llegar a los requisitos de corriente y voltaje pedidos para el sistema.

3.2.5. Etapa 4: Filtrado

El reloj SCLK del potenciómetro está configurado para funcionar a una frecuencia de 10 MHz. Al mirar con el osciloscopio la señal a la salida de las etapas amplificadoras, vemos que la señal que aparece está muy distorsionada por ruido de alta frecuencia. Como después de las etapas amplificadoras se va a sumar la señal a la de los otros n subcircuitos, que van a tener el mismo ruido de alta frecuencia debido a la presencia de un potenciómetro diferente en cada subcircuito, se decide introducir un filtro paso bajo después del último amplificador de cada subcircuito para intentar atenuar el ruido de alta frecuencia. El diagrama de esta red paso bajo es la siguiente:

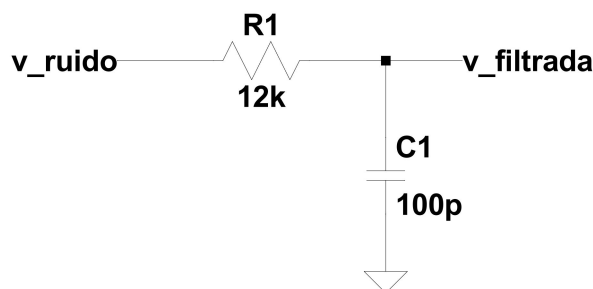


Figure 38. Red paso bajo

La frecuencia de corte que se ha elegido para este circuito es de 132 kHz, para que la banda de paso del filtro incluya de sobra la máxima frecuencia a la que el circuito debe funcionar: 120 kHz. Para conseguir esta frecuencia de corte se han elegido los siguientes valores de resistencia y condensador: 12 kΩ y 100 pF.

$$f_c = \frac{1}{RC} = \frac{1}{2\pi \cdot 12k\Omega \cdot 100pF} = 132.629kHz$$

3.2.6. Etapa 5: Suma

Una vez diseñada la etapa de amplificación y filtrado se probó a acoplar directamente la señal al transistor. Sin embargo, la corriente y el voltaje que se medían en emisor del transistor no eran suficientes para alimentar el LEC ni cumplían los requisitos del sistema.

Para solucionar este problema, se buscó un circuito sumador.

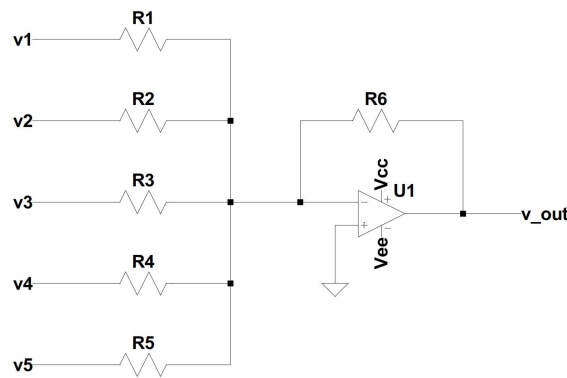


Figure 39. Circuito sumador

Para calcular la tensión de salida, v_{out} de este circuito, debemos primero calcular la corriente que pasa por cada una de las resistencias;

$$i_1 = -\frac{v_1}{R_1}$$

$$i_2 = -\frac{v_2}{R_2}$$

$$i_3 = -\frac{v_3}{R_3}$$

$$i_4 = -\frac{v_4}{R_4}$$

$$i_5 = -\frac{v_5}{R_5}$$

Y por lo tanto i_6 será la suma de cada una de estas corrientes:

$$i_6 = i_1 + i_2 + i_3 + i_4 + i_5$$

$$i_6 = -\left[\frac{v_1}{R_1} + \frac{v_2}{R_2} + \frac{v_3}{R_3} + \frac{v_4}{R_4} + \frac{v_5}{R_5}\right]$$

Y otra parte:

$$v_{out} = i_6 \cdot R_6$$

Sustituyendo la expresión anterior:

$$v_{out} = -R_6 \cdot \left[\frac{v_1}{R_1} + \frac{v_2}{R_2} + \frac{v_3}{R_3} + \frac{v_4}{R_4} + \frac{v_5}{R_5} \right]$$

Para este circuito se han elegido los valores de resistencia:

$$R_1 = R_2 = R_3 = R_4 = R_5 = 10k\Omega$$

$$R_6 = 27k\Omega$$

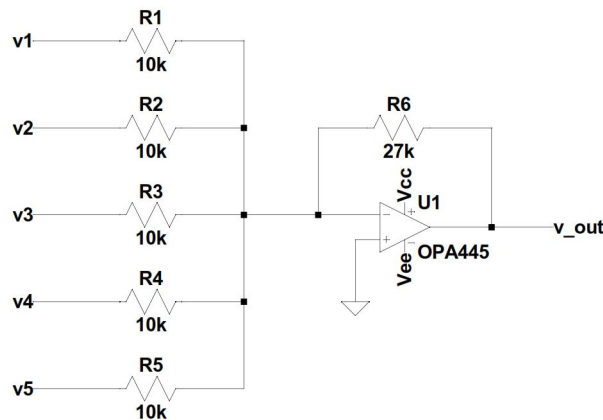


Figure 40. Circuito sumador. Resistencias nominales

Así, la ganancia de esta etapa queda:

$$v_{out} = -27k\Omega \cdot \left[\frac{v_1}{10k\Omega} + \frac{v_2}{10k\Omega} + \frac{v_3}{10k\Omega} + \frac{v_4}{10k\Omega} + \frac{v_5}{10k\Omega} \right]$$

$$v_{out} = -\frac{27k\Omega}{10k\Omega} \cdot [v_1 + v_2 + v_3 + v_4 + v_5]$$

$$v_{out} = -2.7 \cdot [v_1 + v_2 + v_3 + v_4 + v_5]$$

Nótese el signo negativo delante de toda la expresión. Este signo negativo se atribuye a que este sumador es un sumador inversor. Al trabajar con una señal sinusoidal, que es simétrica, no importa que se invierta la misma. Después de probar diferente número de subcircuitos que sumar se optó por sumar 5 subcircuitos. La elección del número de subcircuitos no es para nada arbitraria y viene influenciada por una etapa de switching, una combinación de interruptores que determinarán cuántas señales se suman en esta etapa de suma. El switch analógico que se utiliza tiene 4 switches internos: va a haber siempre una señal que pase por la etapa de suma (de ahí 5-1 = 4 switches). Los switches están controlados por un Arduino y su estado (abierto o cerrado) dependerá de cuánta corriente pida el usuario.

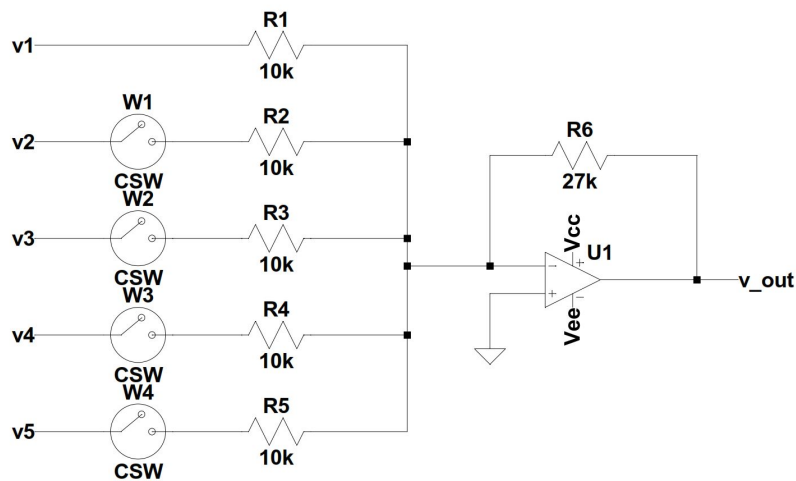


Figure 41. Circuito sumador con interruptores

3.2.7. Etapa 6: Transistor como fuente de corriente

En esta sección se habla de la última etapa del circuito y de dónde se situará el cristal LEC como elemento de carga del circuito.

3.2.7.1. Motivo de esta última etapa

Una de las dudas recurrentes durante el diseño del circuito fue dónde colocar la carga del circuito. Se barajaron diferentes soluciones, como colocar un *buffer* con un amplificador operacional a la salida de la etapa de suma y añadir la carga a la salida del *buffer*, conectándola entre la salida del *buffer* y tierra. Sin embargo, cuando se decidió que era más importante poder monitorizar la corriente que pasase por la carga y que el usuario del sistema pudiese modificar dicha corriente, se optó por añadir un transistor como etapa final. La idea detrás de usar un transistor viene de la necesidad de obtener más corriente con el voltaje que se ha generado en la etapa de suma. Si se monta un circuito seguidor, el voltaje de la base se "seguirá" en el emisor y sin embargo la corriente de emisor (y por consiguiente la de colector) será mucho mayor que la corriente de la base.

Además, si se coloca la carga del circuito en el colector, es decir, el cristal LEC, quedará totalmente aislado del emisor que funciona como fuente de corriente y su adición al circuito no afectará al funcionamiento del mismo.

3.2.7.2. Problemas encontrados durante la implementación de esta etapa

3.2.7.2.1. Elección del transistor

En las primeras iteraciones del diseño del sistema, no se había añadido la etapa de suma y la señal de voltaje en la base del transistor era relativamente pequeña. Esto es, unos 8 V de amplitud, por ejemplo. Por ello, inicialmente se utilizó el transistor NPN BC548. La máxima corriente de colector de este transistor era de 100mA DC y la potencia máxima capaz de disipar era de 625mW.

Además, en cuanto se añadió la etapa de suma, se estaban superando los valores de potencia que podían disipar la resistencia de emisor y el transistor. Una resistencia de emisor baja aumenta la corriente y por tanto la potencia disipada. Llega un momento en el que la potencia disipada es

mayor de la que admite el transistor así que se busca un transistor NPN de potencia, que admita mucha corriente y su potencia máxima disipada tenga un valor elevado. Se decidió que había que cambiar el transistor de inmediato, ya que no paraban de quemarse los BC548. Se eligió el transistor NPN MJE3055T, que es capaz de disipar 75W de potencia y admite corrientes de colector de hasta 10A.

3.2.7.2.2. Elección de la resistencia de emisor

. Durante las primeras iteraciones de diseño del circuito se utilizó como resistencia de emisor del transistor un potenciómetro digital de 100 posiciones y 1 k Ω [19]. La resistencia mínima que este potenciómetro ofrecía eran 10 Ω . 10 Ω es un valor demasiado pequeño para tener en la resistencia de emisor, sobre todo sabiendo los valores de corriente y voltaje que iban a ser generados. Además, se estudió la hoja de datos del potenciómetro que se estaba utilizando y éste sólo admitía una diferencia de voltaje de 8V entre el cursor y uno de los terminales. Sabiendo que el voltaje en el emisor será bastante mayor que 8V, se descartó el uso de este potenciómetro.

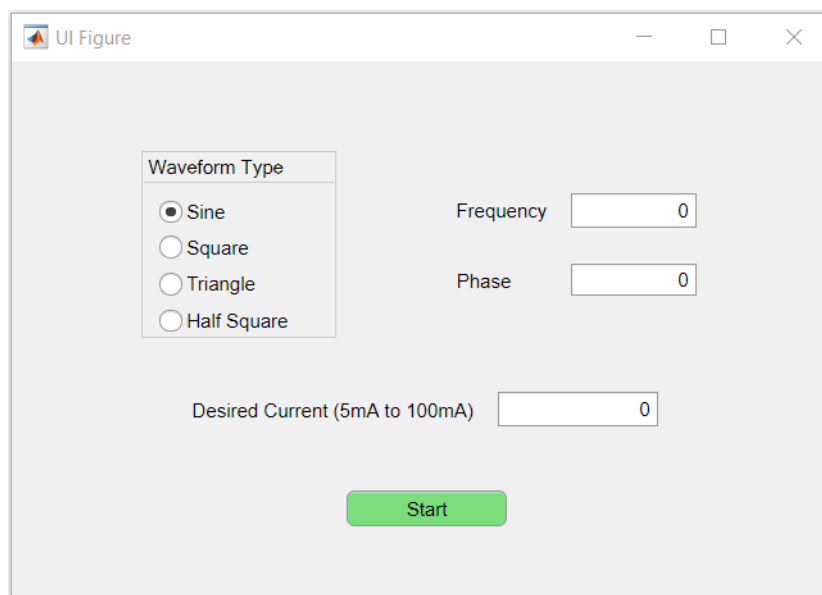
Más tarde, se probó qué resistencia poner utilizando resistencias de 1/4 W de diferentes valores nominales pero se calentaban mucho. Entonces se decidió utilizar un potenciómetro analógico de potencia de 500 Ω y 40W para ver en qué posición del cursor la resistencia era la óptima para el circuito. No se ha encontrado la hoja de datos de este potenciómetro que estaba en el laboratorio. Lo que sí que se sabe es que es un potenciómetro bobinado de 50 Ω y que puede disipar una potencia máxima de 40W. Se eligió entonces 250 Ω como valor de resistencia de emisor.

3.3. Funcionamiento del sistema

El sistema funciona de la siguiente manera:

3.3.1. La interfaz del usuario

El usuario interactúa con el sistema a través de una interfaz de usuario diseñada en MATLAB con el AppDesigner. La interfaz del usuario tiene el siguiente aspecto:



El usuario selecciona entonces el tipo de onda que quiere: sinusoidal, cuadrada, triangular o media cuadrada. También selecciona la frecuencia y la fase que tendrá la señal escribiendo en los cuadros de texto el valor de frecuencia y fase que desee. Estos 3 campos son los que configuran el comando que se enviará al AD9833. Este comando se mandará vía Serial a un Arduino NANO.

El cuadro de texto de "Desired Current (5mA to 100mA)" es el que controla los valores que se van a enviar a los potenciómetros y qué interruptores se van a cerrar dependiendo de cuánta corriente pida el usuario. MATLAB hace cálculos y computaciones con funciones que se han escrito específicamente para este sistema y construye un comando con los valores de las resistencias que debe tomar cada potenciómetro. Este comando se mandará vía Serial a un Arduino NANO.

Por ejemplo, si el usuario quiere que la señal de entrada al circuito sea una onda sinusoidal de 200 Hz de frecuencia y 0 grados de fase, y que la corriente que suministre el circuito a la carga sea de 35 mA, tendrá que seleccionar el botón de "Sine" en el cuadro de Waveform Type, escribir 200 en el cuadro de Frequency, 0 en el cuadro de Phase y 35 en el cuadro de Desired Current. Por último, deberá hacer click en el botón de Start para que ambos comandos se envíen a los correspondientes Arduinos.

La interfaz de usuario presenta un problema. Se deben rellenar los campos de frecuencia, fase o corriente deseada, así como el tipo de onda una segunda vez cuando se arranca el sistema, aún si el valor predeterminado de fase o de tipo de onda (por ejemplo) es el que queremos. Además, hay que cerciorarse de que se pulsa el **botón de Start una única vez**. Esto es importante ya que la conexión de puerto serie con los Arduinos tarda unos segundos en establecerse y otros segundos en enviar el dato. Si se pulsa el botón de Start repetidas veces sin esperar el tiempo necesario para que el puerto serie se abra y se envíen los datos, el programa intentará abrir un puerto que está ocupado abriéndose o enviando, y por lo tanto saltará un error en MATLAB de "puerto ocupado o cerrado". Para volver a tener la interfaz gráfica funcionando después de este error es preciso cerrar el MATLAB por completo y volver a arrancarlo de 0. Sería interesante que en un futuro desarrollo esta aplicación mejorara este aspecto.

3.3.1.1. Generación de comandos para enviar a los Arduinos

Como se ha indicado en los párrafos superiores, el programa de MATLAB construye dos comandos diferentes: uno para enviar el tipo de onda que tiene que generar el AD9833 y otro con los valores de resistencia que deben tener los 5 potenciómetros para suministrar la corriente que pide el usuario.

El primer comando tiene el siguiente formato:

```
cmd1 = 'frecuencia,fase,tipoOnda'
```

El segundo comando tiene el siguiente formato:

```
cmd2 = 'resistencia1,resistencia2,resistencia3,resistencia4,resistencia5'
```

El hecho de que la aplicación se haya programado para generar los comandos dentro de la misma facilita mucho el trabajo del usuario final porque el usuario no tiene que generar ni enviar ningún comando manualmente, todo se realiza dentro de la aplicación, automatizando el proceso.

3.3.1.2. El comando del circuito

El comando que se envía al circuito es cmd2. Envía a los potenciómetros el valor que deben tomar cada vez que se pulse Start en la interfaz del usuario. Esto se realiza calculando cuánta corriente debe suministrar cada canal.

Para calcular cuánta corriente ha de suministrar cada canal, se midió cuánta corriente suministraba un sólo canal, es decir, un subcircuito, a diferentes valores de resistencia del potenciómetro. Esta medición se realizó a mano y cambiando el valor de la resistencia de 100Ω en 100Ω y utilizando el multímetro Agilent para monitorizar la corriente de colector del transistor. Los valores de corriente frente a resistencia se guardaron en un documento de Excel. En muchas ocasiones al cambiar el valor de la resistencia el valor de la corriente no variaba. Por ello, aquellos valores de resistencia en los que no hubiese una variación de corriente no se añadieron al documento de Excel.

Ese documento de Excel se ha convertido en una tabla para ilustrar los valores de corriente frente a resistencia. Además, también se convirtió en un archivo .mat de MATLAB para poder trabajar con ella en el entorno de MATLAB.

Resistencia, Ω	Corriente, mA
100	2.180
550	2.900
950	3.539
1350	4.070
1650	4.073
1700	4.549
2100	5.012
2500	5.443
2900	5.865
3300	6.280
3700	6.692
4100	7.101
4500	7.509
4900	7.916
5300	8.242
5700	8.610
6100	8.909
6500	9.305
6900	9.751
7300	10.216
7700	10.696
8100	11.145
8500	11.576
8900	12.000
9300	12.417
9700	12.827
10100	13.231
10500	13.642
10900	14.049
11300	14.459
11700	14.872

12100	15.287
12500	15.698
12900	15.760
13300	16.200
13700	16.666
14100	17.127
14500	17.564
14900	17.897
15300	18.191
15700	18.469
16100	18.733
16500	18.977
16900	19.213
17300	19.433
17700	19.678
18100	19.893
18500	20.070
18900	20.246
20500	20.273
20900	20.438
21300	20.576
21700	20.811
22100	21.009
22500	21.013
22900	21.111
23300	21.206
23700	21.285
24100	21.374
24500	21.457
24900	21.553
28100	21.580
28500	21.641
28900	21.691
29300	21.746

Table 1. Resistencia vs. Corriente

Como se puede observar en la tabla la corriente mínima que puede suministrar un único subcircuito es de 2.18mA, siendo el valor la resistencia del potenciómetro de 100Ω y la corriente máxima es de 21.746mA, con una resistencia de potenciómetro de $29.3\text{ k}\Omega$. A partir de $29.3\text{ k}\Omega$ la corriente que suministra el subcircuito no incrementa. Para hacer los cálculos más sencillos, se va a aproximar la máxima corriente que suministra cada subcircuito a 22mA.

3.3.1.3. Función de mapeo corriente-resistencia

Para poder utilizar los valores que se presentan en la tabla del apartado superior, se ha escrito una función en MATLAB que asigne cada valor de corriente que pida el usuario con una resistencia determinada. Como los valores de corriente-resistencia se tomaron para un único

subcircuito, la función que se escribió también está orientada al entorno de un subcircuito pero puede ser aplicada a cualquiera de los 5 subcircuitos. El código completo de esta función está en la parte de código fuente de esta memoria.

Evidentemente, la función también debe asignar un valor de resistencia a valores de corriente que no se encuentren en la tabla. Por ejemplo, si el usuario pide un valor de corriente de 15mA, la función tiene que ser capaz de devolver un valor de resistencia que debe de tomar el potenciómetro, ya sea 11700 Ω ó 12100 Ω .

La función *mapCurrentToResistance* hace exactamente eso. Recibe como argumento de entrada un valor arbitrario de corriente proporcionada por el usuario de la aplicación y la misión de esta función es aproximar el valor arbitrario de corriente recibido a uno de los valores de corriente que aparecen en la tabla. Lo asignará al valor más cercano. Esto se consigue acotando el valor arbitrario de corriente entre un entorno pequeño de $\pm\epsilon$. Con un bucle se recorre la matriz de datos entera (la matriz que se ha generado a partir del Excel). Cuando se cumpla que el valor redondeado de la corriente en esa iteración - ϵ es menor que el valor proporcionado por el usuario y el valor proporcionado por el usuario es menor que el valor redondeado de la corriente en esa iteración + ϵ , se ha conseguido acotar el valor. Se podrá pues asignarlo al valor de resistencia ligado a la corriente en dicha iteración en la que se consigue acotar la corriente que pide el usuario.

$$iTabela - \epsilon \leq valorCorrienteArbitraria \leq iTabela + \epsilon \longrightarrow ACOTADO$$

Este proceso se repite dos veces. La primera vez, se declara una $\epsilon_1 = 0.05$ para hacer un barrido más exhaustivo e intentar encontrar el valor de corriente existente en la matriz que más se acerque al valor de corriente que proporciona el usuario. Si con este barrido utilizando ϵ_1 no se ha conseguido acotar el valor, se realiza un segundo barrido con una $\epsilon_2 = 0.3$, un barrido bastante más conservador pero que asegura que cualquier valor de corriente que introduzca el usuario quede mapeado a un valor de resistencia.

Este proceso de acotación y barrido de la matriz no ha sido trivial y se ha tenido que probar diferentes valores de ϵ_1 y ϵ_2 . Para comprobar que los valores de ϵ_1 y ϵ_2 eran los adecuados se realizó un script en MATLAB a modo de banco de pruebas para esta función.

Para el banco de pruebas de la función *mapCurrentToResistance* se declaró un array de valores desde 2.1mA hasta 21.9mA con un paso de 0.05mA. Se aplicó la función *mapCurrentToResistance* a cada uno de los valores del array y se almacenó el resultado en un array de resistencias. Finalmente, se comprobó si algún valor del array de resistencias valía 0. Si existían valores de resistencias igual a 0 significaba que los valores de ϵ_1 y/o ϵ_2 debían ser revisados. Después de varias pruebas con el script del banco de pruebas se consiguieron los valores acertados de ϵ_1 y ϵ_2 y ningún valor del array de resistencias era igual a 0.

```

1  resistanceForRandomCurrent1 = 0;
2  resistanceForRandomCurrent2 = 0;
3  epsilon = 0.3;
4  littleEpsilon = 0.05;
5  roundedValue = round(randomCurrentValue,1); %redondea a la decima mas cercana
6
7  for i = 1:length(currentMatrix(:,1))
8      %first assert if value is closer
9      if((round(currentMatrix(i,2),2)-littleEpsilon) <= randomCurrentValue)
10         if((randomCurrentValue <= round(currentMatrix(i,2),2)+littleEpsilon))
11             resistanceForRandomCurrent1 = currentMatrix(i,1);
12         end
13     end
14
15     if((round(currentMatrix(i,2),1)-epsilon <= roundedValue))
16         if((roundedValue <= round(currentMatrix(i,2),1)+epsilon))
17             resistanceForRandomCurrent2 = currentMatrix(i,1);
18         end
19     end
20 end
21
22 if(randomCurrentValue ~= 0)
23     if((resistanceForRandomCurrent1 < resistanceForRandomCurrent2) &&
24         resistanceForRandomCurrent1 ~= 0)
25         resistance = resistanceForRandomCurrent1;
26     else%(resistanceForRandomCurrent1 == 0)
27         resistance = resistanceForRandomCurrent2;
28     end
29 else
30     resistance = 0;
31 end
32
33 end

```

Código 1. Mapeo corriente-resistencia

3.3.1.4. Funcionamiento del circuito

El circuito funciona gracias al comando que recibe de la aplicación de MATLAB con los valores de resistencia de cada potenciómetro. La aplicación de MATLAB toma el valor que el usuario inserta en el campo de "Desired Current" y llama a una función *generateCommand*, cuyo argumento de entrada es el valor que hay en el campo de "Desired Current" y su argumento de salida es el comando con los valores de las resistencias para enviar al Arduino.

La función calcula las resistencias según qué valor de corriente haya pedido el usuario. En primer lugar, como el circuito siempre va a suministrar corriente, por poca que sea, tiene sentido que siempre haya un subcircuito funcionando. Esto es, la variable *numeroCanales* siempre será mayor o igual que 1. Después se calcula cuántos subcircuitos o canales van a funcionar a su máximo rendimiento, es decir, suministrando 22 mA cada uno. Se calcula de la siguiente manera:

$$nCanalesATope = \left\lfloor \frac{corrienteInput}{maxCorrientePorCanal} \right\rfloor$$

$$nCanalesATope = \left\lfloor \frac{corrienteInput(mA)}{22mA} \right\rfloor$$

Una vez se haya calculado cuántos subcircuitos están funcionando al máximo rendimiento, se comprueba si hay algún subcircuito que tenga que funcionar suministrando una corriente menor de 22mA. Esto se calcula de la siguiente manera:

$$valor = corrienteInput - (nCanalesATope \cdot maxCorrientePorCanal)$$

Si $valor = 0$ significa que todos los subcircuitos que estén funcionando lo harán suministrando 22mA y si $valor \neq 0$ significa que habrá $nCanalesATope$ funcionando al máximo y un subcircuito que suministrará menos corriente. En concreto, suministrará $corrienteInput - (nCanalesATope \cdot 22mA)$.

Una vez realizada esta comprobación podemos asignar su valor final a la variable *numeroCanales*, que inicialmente era valía 1. Si todos los subcircuitos que funcionen lo hacen al máximo rendimiento entonces:

$$numeroCanales = nCanalesATope$$

Y si hay un subcircuito que deba suministrar menos corriente que 22mA:

$$numeroCanales = nCanalesATope + 1$$

Dependiendo del valor que tome *numeroCanales*, se asignarán valores diferentes a cada uno de los potenciómetros llamando a la función *mapCurrentToResistance*. Si $numeroCanales = 1$ significa que sólo habrá un subcircuito funcionando y aportando a la etapa de suma. Por tanto, las resistencias de los potenciómetros son:

```
1 res_pot1 = mapCurrentToResistance(corrienteInput);
2 res_pot2 = mapCurrentToResistance(0);
3 res_pot3 = mapCurrentToResistance(0);
4 res_pot4 = mapCurrentToResistance(0);
5 res_pot5 = mapCurrentToResistance(0);
```

Código 2. Ejemplo de funcionamiento de un sólo subcircuito

Aunque intuitivamente el hecho de que se asigne una corriente igual a 0 a una resistencia (sabiendo que en la tabla de corriente-resistencia no hay corriente 0) vaya a dar error, las funciones escritas en MATLAB tienen controlado este tipo de casos. Cuando el usuario realiza *mapCurrentToResistance(0)*, el valor de resistencia que va a devolver la función es 0, aunque por prudencia tenemos inicializados todos los potenciómetros a 10 kΩ

Cuando $numeroCanales = 3$ ocurre lo siguiente:

```
1 res_pot1 = mapCurrentToResistance(maxCorrientePorCanal);
2 res_pot2 = mapCurrentToResistance(maxCorrientePorCanal);
3 res_pot3 = mapCurrentToResistance(corrienteInput - 2*maxCorrientePorCanal);
4 res_pot4 = mapCurrentToResistance(0);
5 res_pot5 = mapCurrentToResistance(0);
```

Código 3. Ejemplo de funcionamiento de tres subcircuitos

Como se ha explicado anteriormente los canales pueden funcionar a un máximo de corriente. Cuando hay 3 canales funcionando, al menos 2 de ellos estarán suministrando la máxima corriente. El otro puede estar suministrando la máxima corriente o no, pero no hace falta saberlo. El cálculo $corrienteInput - 2 \cdot maxCorrientePorCanal$ da el valor de corriente al que el tercer canal está funcionando, independientemente de su valor.

El resto de casos de los diferentes valores de *numeroCanales* se pueden extrapolar de los dos ejemplos explicados.

Finalmente, cuando se tienen todos los valores de los potenciómetros, se construye el comando que se enviará al Arduino para que lo interprete y aplique los valores de resistencia correspondientes a cada potenciómetro. El comando se construye:

```
1 command = [num2str(res_pot1), ',', num2str(res_pot2), ',', num2str(res_pot3),
2 ',', num2str(res_pot4), ',', num2str(res_pot5)];
```

Código 4. Comando

Durante toda esta sección se ha estado hablando de que puede haber n subcircuitos funcionando y m subcircuitos sin funcionar. Los m subcircuitos que no estén funcionando reciben un valor de resistencia de potenciómetro 0. Pero, ¿cómo sabe el sistema qué subcircuitos tienen que funcionar y cuáles no? Aquí se introduce una etapa intermedia entre el filtrado paso bajo a la salida de la etapa de amplificación y la etapa de suma. Esta etapa intermedia es una etapa que contiene interruptores analógicos. Para esta etapa de interruptores se ha utilizado el ADG444, que contiene 4 interruptores analógicos. Sólo se necesitan 4 interruptores en el circuito porque siempre tiene que haber un subcircuito que esté suministrando corriente, por poca que sea.

Por tanto, si el usuario pide una corriente de 20mA, sólo se necesitará un subcircuito o canal para suministrar esta corriente. Por ello, cuando llamamos a la función *mapCurrentToResistance* en el caso de *numeroCanales* = 1, las resistencias de los potenciómetros 2, 3, 4 y 5 valen 0. El comando que se enviará al Arduino es el siguiente:

$$cmd = [18500, 0, 0, 0, 0]$$

El Sketch que está cargado en el Arduino que recibe este comando calcula cuántas de las 5 resistencias valen 0 y calcula el número de interruptores que tiene que levantar o bajar. En este ejemplo que estamos manejando, el sketch de Arduino tendrá que levantar los 4 switches porque sólo el canal o subcircuito primero está funcionando. Si no tuviésemos esta etapa de interruptores probablemente se perdería precisión en la corriente suministrada al cristal y también habría riesgo de cortocircuitos en las etapas de amplificación al poner resistencia 0 en el potenciómetro.

Si por el contrario tenemos un comando del tipo:

$$cmd = [29300, 29300, 29300, 18900, 0]$$

Sólomente hay un potenciómetro cuya resistencia valdrá 0: el último subcircuito. Con este comando el sketch de Arduino sabe que tiene que cerrar los interruptores 1, 2 y 3 para que se puedan sumar 4 señales en la etapa de suma y así conseguir suministrar al cristal LEC la corriente que pide el usuario.

3.4. Envío de comandos por serial desde MATLAB

La gran mayoría del código de la aplicación de la interfaz del usuario ha sido autogenerado por MATLAB. Lo más relevante de este código es que cuando se pulsa el botón de Start se mandan los comandos a los respectivos Arduinos. Los comandos se envían por la interfaz serial que hay que abrir y cerrar cada vez que se pulse el botón de Start. La sintaxis en MATLAB para declarar objetos serial es:

```
1 %ABRIR LA CONEXIÓN SERIAL:
2 fopen(serial_connection);
3 pause(2); %HAY QUE ESPERAR UNOS SEGUNDOS A QUE SE
4 %ABRA LA CONEXIÓN, DE LO CONTRARIO DA ERROR
5 fprintf(serial_connection, comandoAEnviar);
6 pause(2); %ESPERAR UNOS SEGUNDOS A QUE SE ENVÍEN LOS DATOS
7 %CERRAR LA CONEXIÓN SERIAL:
8 fclose(serial_connection);
```

Código 5. Envío de comandos por interfaz serial

El abrir y cerrar la conexión cada vez que se tenga que enviar un comando es una medida cautelosa de que no se quede el puerto ocupado y no se pueda volver a acceder a él a menos que se cierre completamente el entorno de MATLAB y se vuelva a abrir de nuevo.

Chapter 4

Integración del Sistema

LA etapa final de este TFG es la integración del sistema diseñado en una caja para su uso.

4.1. Soldadura del circuito

Una vez se ha comprobado que el circuito funciona en una protoboard, se procede a realizar una placa soldada con el circuito al completo. El circuito se ha soldado conectando los 5 subcircuítos en paralelo. Uno de los motivos por los que se eligió utilizar Arduino NANO en lugar de Arduino UNO es porque el NANO tiene pines de inserción que permiten soldarlo a una placa. Se pensó que era buena idea ya que ambos Arduinos tienen prácticamente la misma funcionalidad y para nuestro sistema era conveniente tener un circuito lo más integrado y compacto posible.

También, entre otros motivos, se eligió cambiar las tarjetas EVAL-AD9833SDZ y EVAL-SDP-CB1Z por la tarjeta AD9833 ya que tienen la misma funcionalidad y la tarjeta AD9833 es mucho más compacta y tiene pines de inserción para soldarse.

Una vez se decidió cómo iban a distribuirse los componentes a lo largo de la placa se procedió a soldarlos, uno a uno. Hubo que descartar varios de ellos debido a que se quemaron por las altas temperaturas del soldador y tuvieron que ser reemplazados por otros nuevos.

A continuación se adjunta una foto que se hizo cuando el sistema estaba casi totalmente soldado. En esa foto en concreto faltan unos divisores de tensión, el amplificador operacional del sumador y el transistor, pero fueron añadidos más tarde.

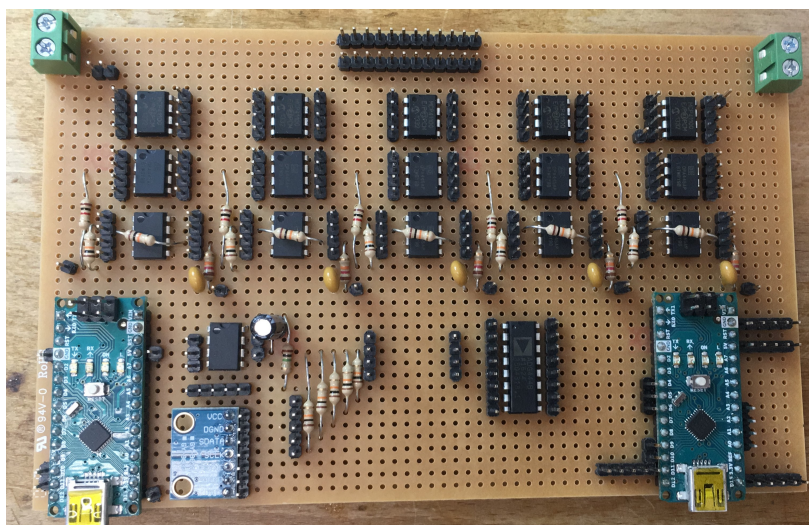


Figure 42. Circuito soldado

El problema principal que presentó esta etapa de soldadura fue el cableado y conexiones entre componentes y etapas del circuito. Se intentó conectar los distintos componentes soldando cables a la placa en las pistas correspondientes. Esto resultó en una maraña de cables que hacía imposible encontrar los problemas de conexiones o *debugging* que el circuito necesitaba. Se soldaron varios componentes con los cables y se alimentó el circuito, probando con el osciloscopio si se veía la señal que se debía ver. Los cables soldados al circuito generaron muchísimos problemas y se optó por desoldar todos los cables y algunos componentes y comenzar la soldadura de nuevo. La solución a este problema se encontró fácilmente soldando pines header macho en las pistas que necesitaban conexión con otro lugar del circuito. Los pines header permiten utilizar cables dupont (en este caso hembra-hembra) para conectar las partes del circuito entre sí fácilmente. Solucionar el problema del cableado de esta manera presentó una ventaja absoluta frente al intento anterior: se podía probar el circuito **componente a componente** simplemente con 10 o 12 cables hembra-hembra conectados al circuito. Así, se comprobó que todos los Arduinos, AD9833, amplificadores operacionales, potenciómetros y switches funcionaban por separado y si había alguno que no funcionase se podía cambiar fácilmente porque no había cables de por medio que impidiesen acceder al componente con facilidad. Para cambiar un componente que se hubiese descubierto estaba dañado durante las pruebas componente a componente, lo único que se debía hacer es quitar los cables hembra-hembra de los pines macho y desoldar con tranquilidad.

Otra ventaja de usar pines macho y cables hembra-hembra fue poder ir comprobando el circuito poco a poco, ir añadiendo etapa tras etapa hasta haber conectado el circuito entero y comprobar su funcionamiento como un sistema integrado.

También se incluyeron un par de filas de varios pines header macho en las alimentaciones de +25V y -25V porque era preciso alimentar muchos componentes a ese voltaje.

Por último, hay que mencionar que la resistencia de potencia de 250Ω y el transistor no se soldaron en la misma placa del resto del circuito. Esta decisión se tomó por dos motivos: el primero, por falta de espacio en la placa para incluir ambos componentes y segundo, porque el transistor y la resistencia se calentaban demasiado (por la gran potencia que debían disipar) como para estar soldados junto a componentes muy sensibles al calor. La resistencia se pegó con cinta de doble cara a un trozo de placa de soldado y se aisló así del resto del circuito. Con el transistor ocurrió algo parecido, solo que al transistor hubo que incluirle un componente más: un disipador de calor. Al igual que la resistencia, se soldó en un trozo de placa externa al circuito junto con el disipador y varios pines header macho para mantener el circuito convenientemente conectado y funcionando. Como trabajo futuro quedará soldar todos los componentes con estaño en una misma placa. También se propone diseñar una PCB con ORCAD de una o varias capas para eliminar el ruido del circuito.

4.2. Integración en una caja

Una vez se comprobó que el circuito estaba correctamente cableado y alimentado y funcionaba bien, era el momento de unificar todas las piezas e introducirlas en una caja para que el sistema fuese funcional y el usuario pudiese usar el sistema de inmediato. Inicialmente se barajó la posibilidad de imprimir una caja con una impresora 3D, pero por falta de tiempo y por las dimensiones del circuito se optó por comprar una caja ya hecha. La caja que se compró tenía forma de paralelepípedo y las caras de menor superficie estaban hechas de plástico para poder realizar agujeros a petición del sistema en cuestión. La caja se muestra a continuación abierta y sin atornillar.



Figure 43. Caja sin montar: tapa con las conexiones

En esta imagen ya se aprecia que a una de las tapas de plástico negro se le han perforado los agujeros correspondientes. Estos agujeros se realizaron con un soldador de gran diámetro. Los agujeros que necesitaba el sistema para poder quedar confinado en una caja eran:

■ Entradas:

- Alimentación +25V.
- Alimentación -25V.
- Alimentación GND o tierra.
- Conexión Arduino 1.
- Conexión Arduino 2.

■ Salidas:

- Ánodo.
- Cátodo.

Para poder conectar cables banana macho a la alimentación de $\pm 25V$, tierra o las salidas de cátodo y ánodo, se han utilizado conectores de banana hembra que quedan en la tapa introducidos por un agujero y ajustados usando una tuerca y una arandela. Para los adaptadores USB-hembra a USB tipo B se han hecho unos agujeros más grandes y finalmente han quedado atornillados a la tapa de la caja.

Una vez terminada la tapa de la caja con todas las conexiones de entradas y salidas se procedió a introducir en el interior de la caja el circuito y colocarlo de manera que dejara espacio para los cables de alimentación y para los cables USB de conexiones de Arduino, sin que quedase nada que se pudiese calentar en contacto con cables o componentes sensibles al calor. La placa soldada con el circuito, la resistencia de potencia y el transistor con el disipador se pegaron al fondo de la caja con cinta de doble cara para asegurar una buena sujeción si la caja se mueve o cae y evitar que se muevan las piezas en el interior de la caja, pudiendo provocar alguna desconexión o fallo en el circuito.

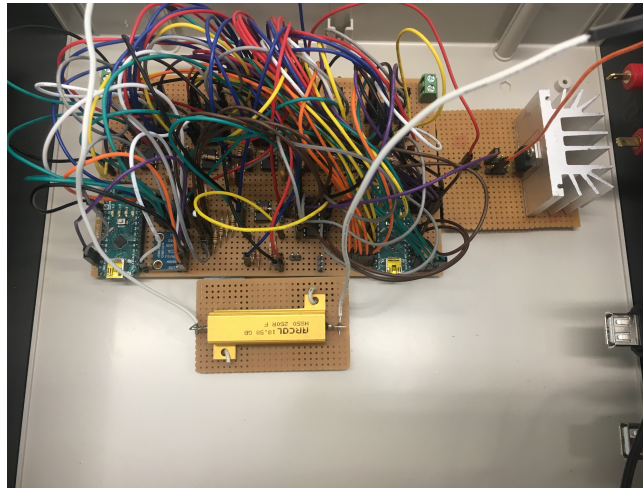


Figure 44. Interior de la caja (sin terminar)

Finalmente los conectores banana hembra se soldaron con cables de alimentación que van introducidos en las clemas soldadas al circuito para poder alimentarlo correctamente y los banana hembra de salida soldados a cables conectados al circuito. Es importante apuntar que la salida queda como un **circuito abierto** entre cátodo y ánodo que se cierra o bien simplemente con un cable banana-banana entre cátodo y ánodo o conectando un componente como una resistencia, un LED o un LEC.

Los cables de USB y alimentación del circuito se pegaron a los costados y fondos de la caja con cinta aislante para evitar contactos con la resistencia o el disipador del transistor y el interior de la caja quedó así:

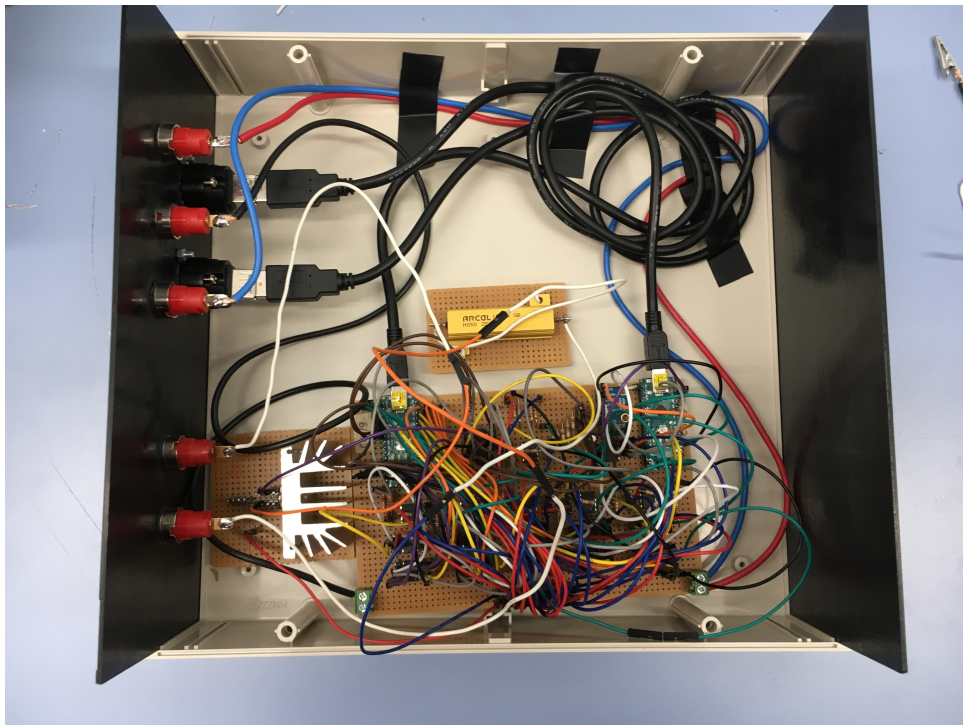


Figure 45. Interior de la caja terminada

Por último, se adjunta una foto de la caja totalmente terminada y cerrada con el cableado correspondiente para que el circuito funcione como debe.

Con esto queda un sistema totalmente funcional integrado en una caja y listo para utilizar.

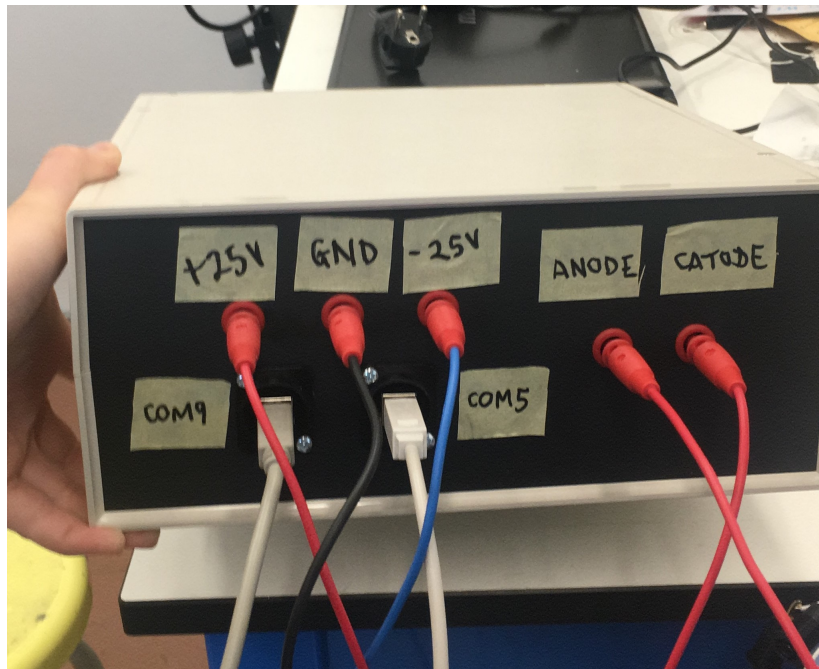


Figure 46. Caja cableada y lista para usar

Chapter 5

Resultados

EN este capítulo se describirán los resultados del sistema, cómo se han conseguido y se hará un estudio del rendimiento del mismo. También se comentarán los problemas encontrados tanto en los resultados como en el diseño del sistema y las soluciones que se han podido implementar o que se pueden ofrecer en un futuro.

5.1. Resultados del sistema

Lo primero que hay que comentar acerca del sistema es si, en efecto, funcionaba con un cristal LEC. Una vez integrado se hicieron pruebas con diferentes cargas: un LED, una resistencia, y un LEC. Efectivamente el LEC se iluminó.

5.1.1. Voltaje a la salida del circuito

El voltaje a la salida del circuito se mide en el emisor, justo antes de la resistencia de emisor. Se ha decidido medirlo aquí en lugar de en el colector ya que para las pruebas de diseño y montaje del sistema se ha trabajado con el transistor a modo de seguidor, sin ninguna carga o resistencia en el colector del transistor. Si se hubiese querido medir la señal del colector sólo se habría podido medir +25V, que es el voltaje al que está el colector durante la etapa de pruebas y medidas. Este voltaje **a máxima ganancia** era demasiado elevado como para poder ver la señal completa en el osciloscopio. Por ello, para observar cuál era la señal a la salida del sistema se optó por trasladar ésta a una protoboard y diseñar un circuito divisor de tensión que atenuase lo suficiente la señal de salida del circuito para poder verla en el osciloscopio. Así, se podría calcular la señal real a la salida utilizando la fórmula del divisor de voltaje. Las resistencias elegidas para el divisor de voltaje son de 82 kΩ y 33 kΩ. Llamaremos a la señal de salida del circuito v_{out} y a la señal que medimos en el divisor de tensión v_{aux} , pues nos ayuda a calcular v_{out} .

La fórmula de éste divisor de tensión es:

$$v_{aux} = v_{out} \cdot \frac{33k\Omega}{33k\Omega + 82k\Omega}$$

Por tanto, para calcular v_{out} teniendo v_{aux} :

$$v_{out} = v_{aux} \cdot \frac{33k\Omega + 82k\Omega}{33k\Omega}$$
$$v_{out} = v_{aux} \cdot \frac{115}{33}$$

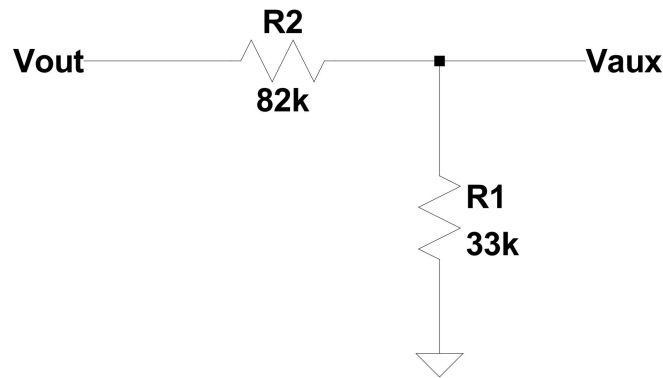


Figure 47. Divisor de voltaje para calcular el voltaje real a la salida

En la siguiente figura se muestra la señal medida en el osciloscopio a la salida del divisor de tensión.

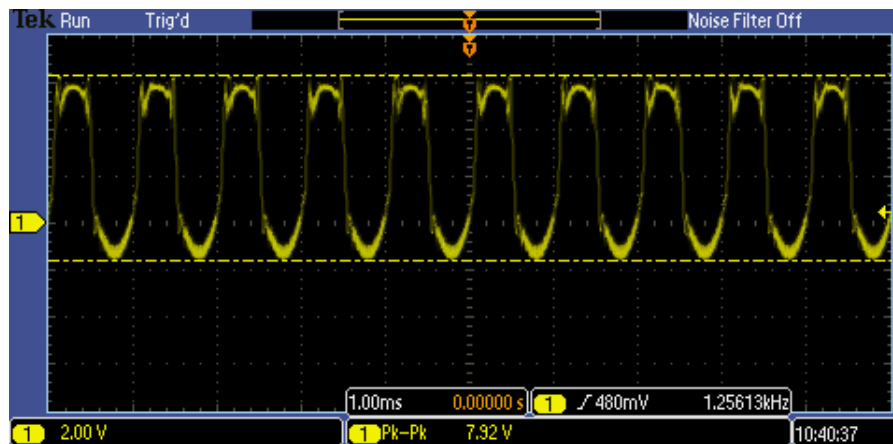


Figure 48. Señal a la salida del divisor auxiliar

Como se puede ver en esta figura, el máximo valor de voltaje de esta señal está en más o menos 6.2 V y el voltaje mínimo de la señal está en -1.6 V, más o menos, lo que efectivamente nos deja con un voltaje pico-pico de 7.92 V. En la figura también se aprecia bastante valor medio en la señal. De hecho, después de los cálculos convenientes se puede ver que la señal a la salida del divisor de voltaje tiene las siguientes características: **2.24 V DC y 3.96 V AC de amplitud.** Por tanto, la señal a la salida del circuito v_{out} , será:

$$2.24 \cdot \frac{115}{33} = 7.806VDC$$

$$3.96 \cdot \frac{115}{33} = 13.8VAC$$

Es decir, **7.806 V de nivel medio y 13.8 V de amplitud de la señal alterna.**

Hay que hacer un pequeño apunte para recordar que la etapa de suma del circuito es en realidad un **sumador inversor**, lo que quiere decir que la señal se invierte. Si el sistema presentase una señal alterna y periódica pura (sin nivel medio) no habría ningún problema, puesto que esa señal invertida sería la señal en sí. Sin embargo, como hay offset en este circuito, quiere decir que al pasar por el sumador la señal queda invertida y el valor medio que antes del sumador era positivo pasará a ser negativo. El motivo de la presencia de este valor medio se ha comentado en capítulos posteriores de la memoria y se comenta una solución para evitar

este offset. Para analizar la señal en el divisor de voltaje mencionado anteriormente se puso un amplificador inversor de ganancia -1 para volver a invertir la señal. Este paso se ha omitido porque no era realmente necesario pero permite estudiar la señal de una manera más intuitiva. En realidad, la señal a la salida del circuito es de **-7.806 V de nivel medio y 13.8 V de amplitud de la señal alterna**.

Por otro lado, sabemos cuál debe ser la señal que debería generar el circuito a ganancia máxima. A ganancia máxima, el circuito debería suministrar 100mA. Con la resistencia de emisor de $250\ \Omega$, tenemos que la amplitud de la señal de salida debe ser:

$$v_E = 250\Omega \cdot 100mA = 25V$$

Lo que significa que **la señal de salida debería ser de 25V**.

5.1.1.1. Rendimiento en términos de voltaje

Para calcular el rendimiento en términos de voltaje, se ha utilizado MATLAB para mostrar la señal medida en el osciloscopio y la señal teórica, ya que es más intuitivo ver una imagen.

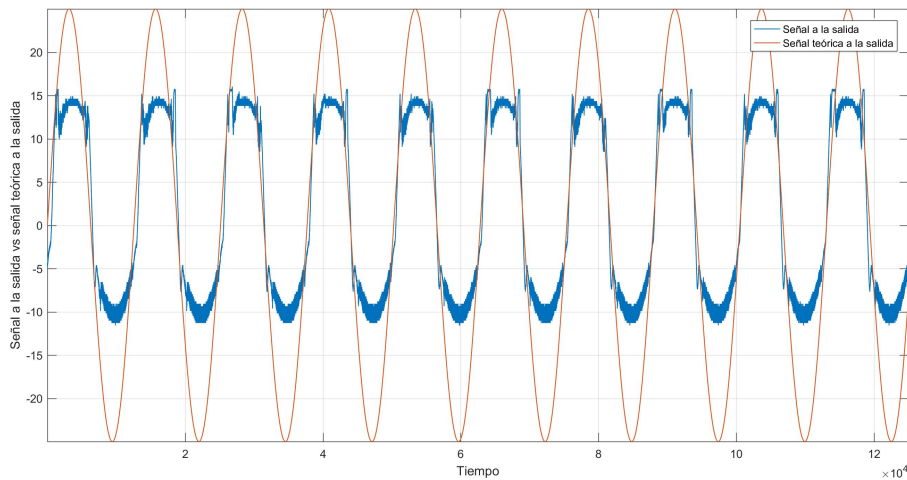


Figure 49. Señales a la salida: real vs. teórica

Calculamos el rendimiento como el cociente entre los voltajes pico-pico de las dos señales:

$$\eta = \frac{V_{pp,señalreal}}{V_{pp,señalteórica}} = \frac{22.0242 - (-5.5758)}{25 - (-25)} = \frac{27.6}{50} = 55.2\%$$

Por tanto, el rendimiento del circuito en términos del voltaje de salida es: $\eta = 55.2\%$. La causa de este bajo rendimiento se comenta en la sección 5.2.1.2.

5.1.2. Corriente pedida vs. Corriente suministrada

En esta sección se ilustra la corriente pedida por el usuario desde la aplicación y la corriente suministrada por el circuito. Para realizar estas mediciones se ha conectado el Multímetro Agilent en el colector del transistor para así medir correctamente la corriente AC. Las medidas se han tomado una a una variando la corriente de 0.5 mA en 0.5 mA pedida en la interfaz del usuario del sistema. La tabla es la siguiente:

Pedida, mA	Suministrada, mA
5.00 0	5.220
5.50 0	5.651
6.00 0	6.548
6.50 0	6.980
7.00 0	7.407
7.500	7.841
8.000	8.706
8.500	9.143
9.000	9.611
9.500	10.255
10.000	10.906
10.500	11.454
11.000	11.972
11.500	12.481
12.000	12.960
12.500	13.902
13.000	14.362
13.500	14.827
14.000	15.299
14.500	15.751
15.000	16.656
15.500	16.573
16.000	17.232
16.500	17.893
17.000	18.457
17.500	18.982
18.000	19.984
18.500	20.456
19.000	21.421
19.500	22.670
20.000	23.725
20.500	24.061
21.000	24.274
21.500	24.722
22.000	24.851
22.500	24.879
23.000	24.890
23.500	24.891
24.000	27.206
24.500	27.203
25.000	44.410
25.500	44.495
26.000	44.506
26.500	28.879
27.000	28.983
27.500	29.228
28.000	30.019

28.500	30.642
29.000	31.327
29.500	31.912
30.000	32.908
30.500	33.414
31.000	34.059
31.500	44.856
32.000	44.888
32.500	44.891
33.000	37.481
33.500	37.682
34.000	38.160
34.500	39.226
35.000	39.704
35.500	40.186
36.000	40.627
36.500	41.086
37.000	40.293
37.500	38.776
38.000	44.875
38.500	44.903
39.000	44.904
39.500	38.043
40.000	37.785
40.500	37.673
41.000	37.182
41.500	36.383
42.000	44.856
42.500	34.605
43.000	34.246
43.500	33.662
44.000	34.183
44.500	34.252
45.000	34.311
45.500	34.363
46.000	36.354
46.500	36.380
47.000	37.032
47.500	37.052
48.000	37.351
48.500	37.805
49.000	37.502
49.500	37.587
50.000	38.239
50.500	38.606
51.000	39.291
51.500	40.354

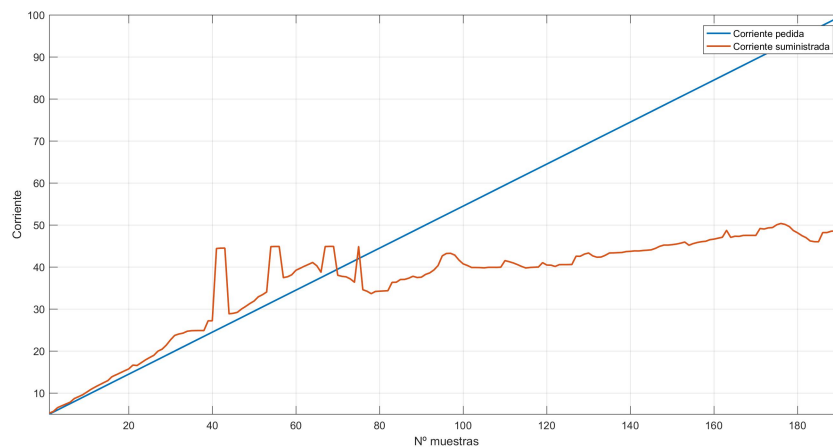
52.000	42.658
52.500	43.225
53.000	43.265
53.500	42.836
54.000	41.654
54.500	40.752
55.000	40.372
55.500	39.891
56.000	39.888
56.500	39.872
57.000	39.806
57.500	39.925
58.000	39.932
58.500	39.920
59.000	39.965
59.500	41.526
60.000	41.274
60.500	40.968
61.000	40.576
61.500	40.128
62.000	39.782
62.500	39.900
63.000	39.953
63.500	40.000
64.000	41.041
64.500	40.473
65.000	40.445
65.500	40.139
66.000	40.569
66.500	40.577
67.000	40.582
67.500	40.600
68.000	42.570
68.500	42.561
69.000	43.081
69.500	43.337
70.000	42.649
70.500	42.337
71.000	42.382
71.500	42.795
72.000	43.346
72.500	43.368
73.000	43.423
73.500	43.462
74.000	43.680
74.500	43.732
75.000	43.843

75.500	43.826
76.000	43.932
76.500	44.000
77.000	44.095
77.500	44.451
78.000	44.925
78.500	45.220
79.000	45.222
79.500	45.326
80.000	45.462
80.500	45.675
81.000	45.943
81.500	45.189
82.000	45.569
82.500	45.840
83.000	46.031
83.500	46.125
84.000	46.503
84.500	46.643
85.000	46.871
85.500	47.087
86.000	48.719
86.500	47.056
87.000	47.326
87.500	47.292
88.000	47.524
88.500	47.528
89.000	47.533
89.500	47.521
90.000	49.163
90.500	49.048
91.000	49.318
91.500	49.405
92.000	50.047
92.500	50.367
93.000	50.137
93.500	49.639
94.000	48.666
94.500	48.131
95.000	47.482
95.500	47.000
96.000	46.206
96.500	46.030
97.000	46.020
97.500	48.196
98.000	48.200
98.500	48.513

99.000	48.602
99.500	48.410
100.000	48.288

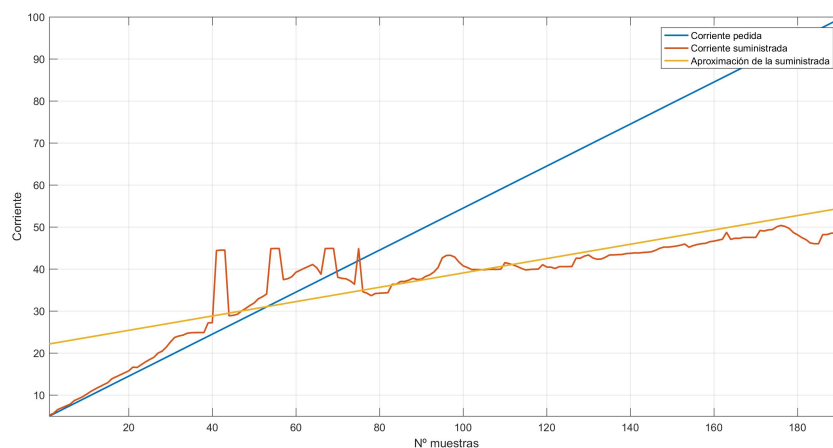
Table 2. Corriente pedida vs. Corriente suministrada

La siguiente figura generada en MATLAB ilustra los datos de la tabla:

**Figure 50.** Corrientes a la salida: pedida vs. suministrada

5.1.2.1. Rendimiento en términos de corriente

En cuanto al rendimiento en términos de corriente, también se ha utilizado MATLAB para hacer su cálculo. Como se ha visto en la figura de la sección anterior, en términos de corriente el sistema tiene un rendimiento bastante pobre y además la corriente suministrada muestra algunas anomalías en ciertos valores de corriente pedida. Sin embargo, se puede ver que a niveles bajos de corriente el sistema funciona bastante bien.

**Figure 51.** Corrientes a la salida: pedida vs. suministrada y aproximaciones

Para analizar el rendimiento en términos de corriente se ha separado la gráfica en dos partes: antes y después de las anomalías de corriente. Antes de las anomalías, es decir, a niveles bajos

de corriente, podemos observar que el circuito se comporta bastante bien. De hecho, incluso suministra un poco más de corriente de la pedida. Sin embargo, después de las anomalías el sistema pierde mucha calidad. La corriente suministrada se ha aproximado a una recta para hacer los cálculos de rendimiento más sencillos, la gráfica amarilla. El rendimiento del circuito a niveles altos de corriente es:

$$\eta = \frac{\text{corrienteSuministrada}_{\text{aprox}}}{\text{corrientePedida}} = 64.48\%$$

5.2. Problemas encontrados

Como todo en esta vida, ha habido problemas. En esta sección se va a proceder a enunciar los problemas que ha habido durante el proceso de diseño e implementación del circuito y cuál es la posible causa de estos inconvenientes.

5.2.1. Anomalías del potenciómetro MCP41100

Ya se ha indicado en secciones anteriores, pero es bueno comentarlo de nuevo: el potenciómetro digital MCP41100 es un componente que no admite altas potencias y su funcionamiento no es el óptimo. A continuación se enuncian algunos de los problemas que ha generado este componente electrónico.

5.2.1.1. Anomalías en resistencia

Como se puede ver en la figura anterior, la corriente suministrada en ciertas ocasiones presenta anomalías, suministrando mucha más corriente de la pedida durante 5 o 6 muestras. Por ejemplo, cuando se piden 25mA el sistema suministra 44.41mA, pero en muestras anteriores y posteriores el circuito parece funcionar correctamente. Estas anomalías ya se presentaron durante la etapa de familiarización con el potenciómetro digital, pero no se han conseguido solucionar y no se ha tenido tiempo para investigar el motivo por el cual el potenciómetro tiene este comportamiento.

5.2.1.2. *Soft Clipping* o recorte de la señal

El *soft clipping* es un fenómeno que ocurre cuando la señal se recorta de manera suave, al contrario del *hard clipping*, que ocurre cuando la señal presenta un recorte brusco, como por ejemplo las señales que quedan saturadas a la salida de un amplificador operacional cuando superan el voltaje de alimentación del operacional. La señal a la salida de la primera etapa de amplificación presenta *soft clipping* a partir de cierto valor de ganancia, esto es, a partir de cierto valor de resistencia enviado al potenciómetro.

En un estudio posterior a la implementación, con ánimo de encontrar cuál era el motivo por el cual ocurría el *soft clipping* se revisó la hoja de datos del potenciómetro. Se encontró que los valores máximos de tensión que admiten los terminales del potenciómetro son iguales a la alimentación que recibe el potenciómetro. En este caso, los 5V de alimentación que suministra el Arduino. Sin embargo, en el diseño que está funcionando en estos momentos hay un valor de resistencia del potenciómetro a partir del cual la tensión en sus terminales es mayor que 5V. Una posible solución a éste problema se discute en el capítulo 6. La explicación de por qué ocurre esto es la siguiente:

La tensión de los terminales de un amplificador operacional debe ser la misma y por tanto la diferencia de voltaje entre los terminales es más o menos 0. En la primera etapa de amplificación,

la cual tiene ganancia variable, la tensión en el terminal positivo es de 0.33V. Entonces, la tensión en el terminal negativo será de 0.33V también. Esto significa que por la resistencia de $1k\Omega$ pasa la siguiente corriente:

$$i = \frac{0.33V}{1k\Omega} = 33\mu A$$

Que es la misma corriente que por lo tanto pasará por los terminales del potenciómetro. Se supera el umbral de los 5V en los terminales del potenciómetro cuando la resistencia del mismo supera el valor:

$$R_{pot} = \frac{5V}{33\mu A} = 15.151k\Omega$$

Las siguientes figuras son capturas del osciloscopio. Las medidas están realizadas a la salida de la etapa de amplificación. La primera captura muestra la señal amplificada con recorte en prácticamente toda la zona negativa de la señal sinusoidal. La resistencia del potenciómetro en este caso es de $5k\Omega$. En la segunda figura se muestra el mismo caso pero la resistencia del potenciómetro vale $23.9k\Omega$. En este caso se puede observar que el recorte ocurre en toda la parte negativa de la señal y en las crestas positivas de la señal.

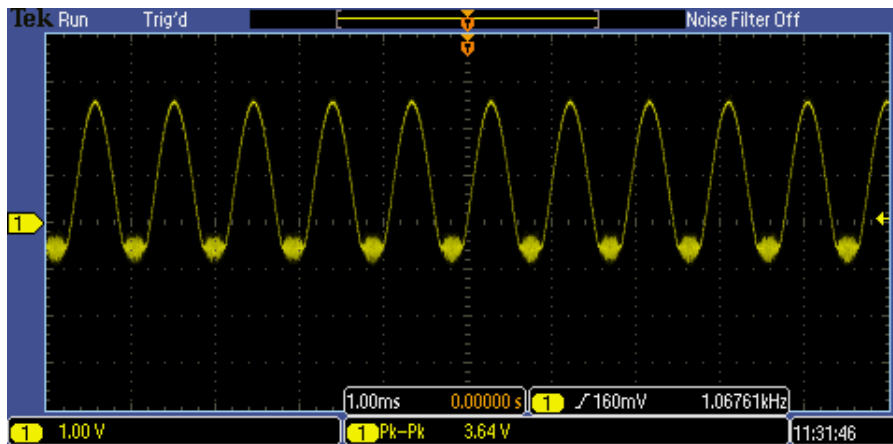


Figure 52. Soft Clipping a $R_{pot} = 5k\Omega$

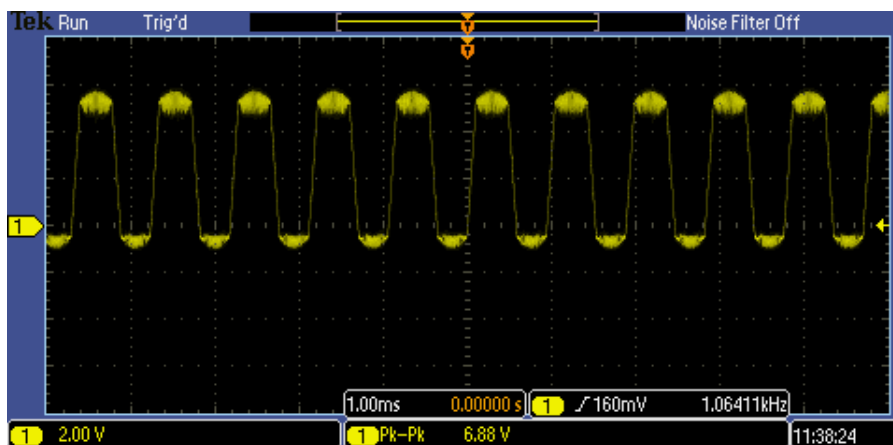


Figure 53. Soft Clipping a $R_{pot} = 23.9k\Omega$

No se tiene la certeza absoluta de que el hecho de haber superado el valor máximo de tensión permitida entre los terminales del potenciómetro genere el *soft clipping*, pero no se ha encontrado

otra posible causa. Este es el motivo principal del bajo rendimiento de la señal en términos de voltaje y corriente.

5.2.1.3. Ruido de alta frecuencia

La hoja de datos del potenciómetro MCP41100 indica que a una tensión de alimentación $V_{DD} = 5V$ la frecuencia de reloj del SPI es de 10MHz. Si se observan las figuras de la sección anterior se puede apreciar un ruido de alta frecuencia bastante considerable en las zonas en las que hay *clipping*. Se sospecha que este ruido viene causado por el reloj del SPI del potenciómetro, que de alguna manera interfiere con el resto del circuito inyectando este ruido. Una solución que se ha implementado al circuito se discute en la sección 5.3.1.

5.2.2. Nivel medio

Se puede observar claramente que en presencia de *soft clipping* la señal, al perder parte de ella por el recorte, presenta un aspecto muy diferente al esperado y además el recorte genera un **nivel medio** que no se esperaba tener al diseñar el circuito. Este nivel medio se ha calculado y discutido en este capítulo y su valor es de 7.806 V. Es importante mencionar la presencia del nivel medio ya que, al hacer pruebas con un LED de potencia, se observaba que el LED se iluminaba bastante aún pidiendo desde la interfaz gráfica que suministrase poca corriente (por ejemplo 15mA). Para concluir que este brillo lo causaba el nivel medio se redujo la frecuencia de la señal de entrada al circuito a 5 Hz, para que el parpadeo fuese visible al ojo humano. Efectivamente, se notaba un parpadeo sobre un brillo constante. El parpadeo, en la etapa de "apagado" tenía el brillo constante mencionado anteriormente y en la etapa de "encendido" el brillo era mucho mayor. Esto puede afectar gravemente a la medida de impedancia de los LEC en el caso de que este sistema se vaya a utilizar.

5.3. Soluciones

En esta sección se discuten las posibles soluciones a los problemas encontrados en el sistema.

5.3.1. Ruido de alta frecuencia: filtro paso bajo

Tal y como se ha explicado en la sección 3.2.5: Filtrado, después de las etapas de amplificación de los subcircuitos se incluye una red paso bajo. Como se sabe a qué rango de frecuencias debe funcionar el circuito y que la frecuencia del reloj es de 10MHz, se ha decidido que el filtro paso bajo tenga una frecuencia de corte de 132kHz, lo cual se consigue con una resistencia de $R = 12k\Omega$ y un condensador de $C = 100pF$.

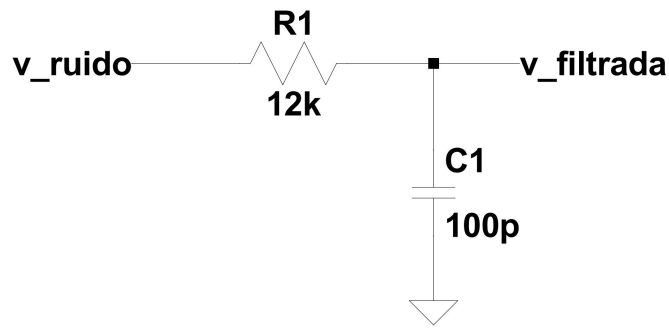
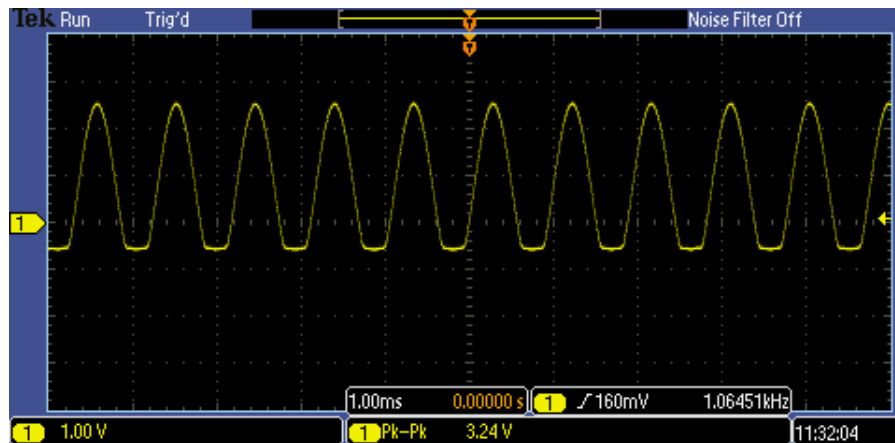
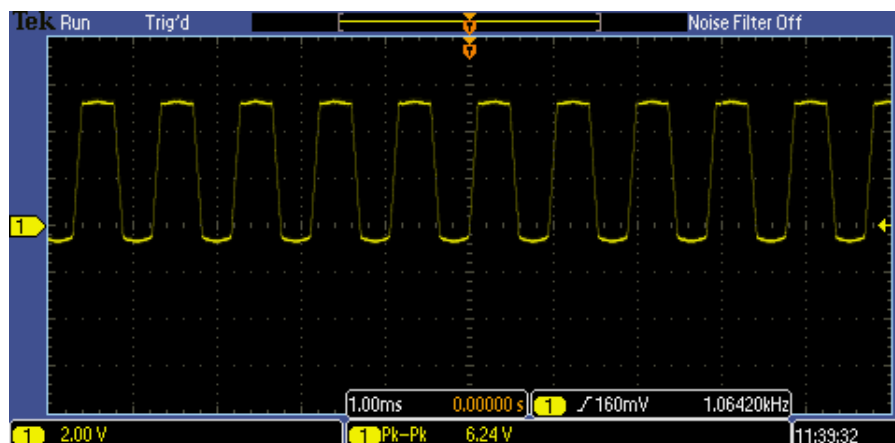


Figure 54. Circuito filtro paso bajo

Las figuras que se muestran a continuación son las señales de la sección anterior medidas justo después de la etapa de filtrado. Se puede observar que, efectivamente, se elimina el ruido dejando una señal mucho más limpia y fácil de comprender.

Figure 55. Soft clipping solucionado para $R_{pot} = 5k\Omega$ Figure 56. Soft clipping solucionado para $R_{pot} = 23.9k\Omega$

5.3.2. ¿Solución al *Soft Clipping*?

Durante la redacción de ésta memoria se encontró un posible motivo del *soft clipping* que presentaba el circuito. Por otro lado, no se ha montado un circuito alternativo mejorado, pero sí

que se ha diseñado una propuesta que simplifica y abarata el sistema. Esta nueva propuesta se discute en el capítulo siguiente.

Chapter 6

Conclusiones y futuros desarrollos

COMO cierre a este trabajo de fin de grado se discutirán ciertas conclusiones que se han sacado después de todo el diseño del sistema y redacción de la memoria.

6.1. Conclusiones

Se recordarán brevemente los requisitos del sistema. El sistema debe suministrar una corriente alterna de 5mA a 100mA y se debe inyectar al cristal LEC un voltaje mayor de 20V. El sistema debe tener un barrido de frecuencias de 5Hz hasta 120kHz. Por último, debe crearse una interfaz amigable con la que el usuario pueda interactuar y configurar el tipo de onda y cuánta corriente se quiere suministrar al cristal LEC. Los objetivos iniciales del sistema eran poder generar una señal con frecuencias de hasta MHz y poder medir la impedancia en la parte OFF de la señal pulsada o sus niveles bajos. Por falta de tiempo el trabajo se ha centrado en los requisitos comentados al principio de este párrafo.

Uno de los temas a discutir en este capítulo es el rendimiento del sistema. En términos de corriente, el sistema tiene un rendimiento del 64.48%. Ya se ha comentado en capítulos anteriores el motivo de este rendimiento. No es un rendimiento bueno, evidentemente. Hay muchos factores afectan a que éste valor no sea del 100%. Por ejemplo, utilizar componentes de potencia que disipan mucho calor afecta al rendimiento del sistema, el uso no óptimo de los potenciómetros MCP41100, etc.

También hay que señalar que este TFG ha tenido restricciones de tiempo, ya que se ha realizado en algo menos de 3 meses. Durante la redacción de esta memoria se ha entendido cómo implementar un mejor sistema: más sencillo, elegante e igual de funcional que el que se ha implementado en estos últimos dos meses. De haber tenido más tiempo para realizar este trabajo probablemente se hubiese llegado a la solución propuesta en la siguiente sección de manera natural y se podría haber probado su eficacia y rendimiento.

En este trabajo de fin de grado se han aplicado todos conceptos de electrónica aprendidos durante el grado. Se ha aplicado la capacidad de analizar un problema, capacidad de sintetizar y encontrar una solución al problema planteado. Se ha intentado facilitar el trabajo al usuario final realizando una interfaz gráfica intuitiva y fácil de utilizar y minimizando el número de entradas y salidas que necesite el sistema, para que el usuario no necesite configurar nada externo a la propia interfaz gráfica.

Por último se quiere comentar que el sistema se ha probado con un cristal LEC y se ha visto que el cristal se ilumina gracias a la potencia que suministra el sistema. Se recomienda que antes

de intentar duplicar el sistema descrito en este TFG se intente implementar el diseño que se ha propuesto en la continuación y evaluar el rendimiento del circuito propuesto, para ver si merece la pena duplicar este sistema o sustituir el circuito por la nueva propuesta.

6.2. Mejoras y futuros desarrollos

Como siempre hay que ser crítico con uno mismo y admitir sus errores, en este capítulo se ofrece también una posible mejora al sistema diseñado. La mejora se basa principalmente en el diseño de un nuevo circuito, basándose en los mismos requisitos y utilizando los mismos componentes. Este es el esquema de la propuesta:

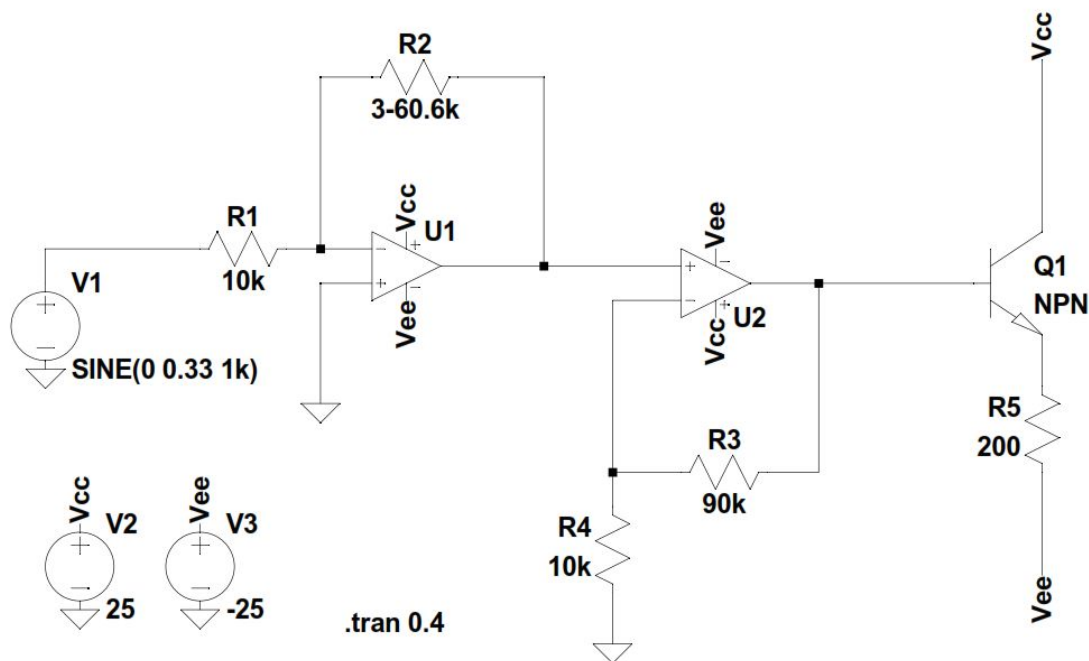


Figure 57. Diseño planteado

En este diseño se han obviado las dos etapas de filtrado y se ha puesto el foco de atención en la etapa de amplificación. El ánimo de diseñar un circuito nuevo tiene varias motivaciones:

- De mejorar el rendimiento del sistema y poder suministrar la corriente pedida.
- Del intento de eliminar el *soft clipping* que el sistema descrito en este trabajo presenta.
- De conseguir un funcionamiento óptimo del sistema.
- De buscar una solución sencilla y elegante, sin complejidad añadida y con la funcionalidad esencial para que el sistema cumpla los requisitos pedidos.

Esta propuesta tiene también dos etapas de amplificación. A diferencia del implementado en este TFG, la primera etapa de amplificación es inversora. Además, este sería el circuito completo. No habría necesidad de montar otros cuatro subcircuitos diferentes ni de añadir un interruptor analógico a cada subcircuito. Tampoco resultaría necesaria la etapa de suma, ya que no hay más subcircuitos que haya que sumar. Toda la corriente es posible suministrarla con un sólo circuito.

6.3. Análisis del circuito

Para diseñar este circuito se comenzó recogiendo los requisitos que debe tener el sistema en términos de corriente: que suministre de 5mA a 100mA de corriente alterna. Esto se conseguirá, igual que en el circuito descrito en el capítulo 3, con un transistor. En este caso se ha elegido como valor nominal de la resistencia de emisor 200Ω .

En el modelo de pequeña señal de la etapa final del circuito (transistor como fuente de corriente), la corriente de colector se calcula:

$$i_C = i_E = -\frac{v_E}{R_E} = -\frac{v_E}{200}$$

La corriente mínima que debe suministrar el circuito es de **5 mA**:

$$5mA = \frac{v_E}{200}$$

$$v_E = 1V$$

Y, además, en el modelo de pequeña señal no hay caída de tensión v_{BE} , por lo que $v_B = v_E = 1V$. Con este dato ya podemos calcular la ganancia mínima de la etapa amplificadora.

Por otro lado, la corriente máxima que debe suministrar el circuito es de **100mA**:

$$100mA = \frac{v_E}{200}; v_E = 20V$$

$$v_E = 20V$$

Y $v_B = 20V$ con lo que también tenemos la ganancia máxima de la etapa amplificadora.

La ganancia se define como:

$$G = \frac{v_{out}}{v_{in}}$$

Y por ello:

$$G_{max} = \frac{-20V}{0.33V} = -60.6\left(\frac{V}{V}\right)$$

$$G_{min} = \frac{-1V}{0.33V} = -3.03\left(\frac{V}{V}\right)$$

La primera etapa de amplificación es de ganancia variable, al igual que en el sistema implementado. Sin embargo, se trata de un amplificador inversor. La fórmula de la ganancia para esta etapa es:

$$G_{var} = -\frac{R_{pot}}{10k\Omega}$$

La segunda etapa de amplificación es una etapa de amplificación no inversora de ganancia fija $G = 10(V/V)$.

$$G_{fija} = 1 + \frac{90k\Omega}{10k\Omega} = 10\left(\frac{V}{V}\right)$$

La ganancia total del circuito es de:

$$G_{total} = G_{var} \cdot G_{fija}$$

$$G_{total} = \left(-\frac{R_{pot}}{10k\Omega}\right) \cdot 10$$

Para saber cuáles son los valores que debe tomar la resistencia del potenciómetro en los casos extremos, es decir, en los de ganancia máxima y mínima, solo necesitamos despejar de la ecuación anterior R_{pot} .

$$R_{pot} = -\frac{G_{total}}{10} \cdot 10k\Omega$$

Lo que queda:

$$R_{pot} = -G_{total} \cdot 1000$$

Para la ganancia máxima G_{max} el potenciómetro debe tener la siguiente resistencia:

$$R_{pot,max} = -G_{max} \cdot 1000$$

$$R_{pot,max} = -(-60.6) \cdot 1000$$

$$R_{pot,max} = 60.6k\Omega$$

Para la ganancia mínima G_{min} el potenciómetro debe tener la siguiente resistencia:

$$R_{pot,min} = -G_{min} \cdot 1000$$

$$R_{pot,min} = -(-3.03) \cdot 1000$$

$$R_{pot,min} = 3.03k\Omega$$

Esto es, **el valor del potenciómetro debe estar en el rango de 3.03 kΩ a 60.6 kΩ** para un correcto funcionamiento del circuito. El superar el valor de 60.6 kΩ supondrá un aumento de la ganancia de la etapa amplificadora, lo que significa que la señal de voltaje de salida tendrá más amplitud y la corriente de colector máxima tendrá un valor mayor que 100 mA.

Este circuito tiene dos características interesantes:

- Permite un ajuste fino de corriente pedida por el usuario al tener una etapa inversora cuya ganancia puede ser menor que 1 (cosa que no ocurre en los amplificadores no inversores).
- Limita los valores de corriente que atraviesan el potenciómetro, discutidos en el capítulo anterior y evita que el voltaje en los terminales del potenciómetro sea mayor que la tensión de alimentación del Arduino (5V). Quedaría por probar que efectivamente esta hipótesis es cierta y que los potenciómetros funcionan a esta potencia.

Bibliography

- [1] **Pei Q, Y. G.** *Polymer light-emitting electrochemical cells*. 1995.
- [2] **Ruben D. Costa, E. O.** *Luminescent Ionic Transition-Metal Complexes for Light-Emitting Electrochemical Cells*. *Angewandte Reviews*. 2012.
- [3] **Arduino.** *Arduino NANO Guide*.
Retrieved from:
<https://www.arduino.cc/en/Guide/ArduinoNano>
- [4] **Analog Devices (2011-2012).** *Evaluation Board User Guide. UG-272. Evaluating the AD9833 Low Power 12.65 mW, 2.3 V to 5.5 V, Programmable Waveform Generator*.
Retrieved from Analog Devices:
<https://www.analog.com/media/en/technical-documentation/user-guides/UG-272.pdf>
- [5] *Agilent Multimeter Datasheet*. Retrieved from:
<https://d3fdwrtpsindh7j.cloudfront.net/Docs/datasheet/34401A.pdf>
- [6] **Microchip.** *Microchip MCP41100*. Retrieved from:
<http://ww1.microchip.com/downloads/en/devicedoc/11195c.pdf>
- [7] **Texas Instruments.** *Amplificador operacional OPA445*. Retrieved from:
<https://www.ti.com/lit/ds/symlink/opa445.pdf>
- [8] *Transistor NPN MJE3055*. Retrieved from:
<https://www.mouser.com/ds/2/149/MJE3055T-889641.pdf>
- [9] **Analog Devices.** *Switch ADG444*. Retrieved from:
https://www.analog.com/media/en/technical-documentation/data-sheets/ADG441_442_444.pdf
- [10] **Arcol resistors.** *50 Ω Arcol resistor*.
Retrieved from:
<http://www.arcolresistors.com/wp-content/uploads/2014/03/HS-Datasheet.pdf>
- [11] **Chia-Yu Cheng and Hai-Ching Su.** *Light-Emitting Electrochemical Cells*. Julio 2016.
Retrieved from Research Gate:
<https://www.researchgate.net/publication/305708257>
- [12] **Antonio Pertegás Ojeda.** *Highly Efficient Light-Emitting Electrochemical Cells*.
Noviembre 2015.
- [13] **Andreas Sandstrom.** *Design and Fabrication of LightEmitting Electrochemical Cells*.
2013.
Retrieved from: <http://umu.diva-portal.org/>

- [14] **Thomas C. Hayes, Paul Horowitz.** *Student Manual for the Art of Electronics*. Julio 1989.
- [15] **Paul Horowitz.** *The Art Of Electronics - 2nd Edition*. 1989.
- [16] *Amplificador sumador*. Retrieved from:
<http://hyperphysics.phy-astr.gsu.edu/hbasees/Electronic/opampvar5.html>
- [17] **GitHub**, *Librería de código para el AD9833*.
Retrieved from:
<https://github.com/Billwilliams1952/AD9833-Library-Arduino>
- [18] **GitHub**, *Librería de código para los potenciómetros MCP41XXX*.
Retrieved from:
https://github.com/sleemanj/MCP41_Simple
- [19] **Renesas.** *Potenciómetro X9C102*. Retrieved from:
<https://www.renesas.com/eu/en/www/doc/datasheet/x9c102-103-104-503.pdf>

PART II



CÓDIGO FUENTE



Chapter 1

Arduino

1.1. Código de programación del AD9833

```
1 /* AD9833_MATLAB
2  * @author Paula de Vega Botas
3  * July 25, 2019
4  *
5  * ##### IMPORTANT INFORMATION #####
6  *
7  * This program must be loaded to the Arduino NANO
8  * that is on the left side of the circuit board.
9  * If you're lost with this indication, look for
10 * the +25V and -25V input of the whole circuit board.
11 * That's the side you're looking for.
12 *
13 * When working in the computer at IMDEA Arduino IDE
14 * recognised this Arduino NANO board in Port COM5,
15 * but it depends on which computer this program is
16 * going to be compiled and opened, so pay attention
17 * at which port recognises the Arduino NANO board
18 * when using another computer. Just keep in mind
19 * there are two Arduino NANO boards in this circuit
20 * and it is very important that each NANO gets the
21 * right sketch loaded. Otherwise the system will not work.
22 *
23 * This Arduino NANO controls the signal generator so
24 * be very careful while uploading the sketch to
25 * the Arduino board. Please turn off the voltage supply
26 * and again make sure you're loading the right
27 * sketch into the right Arduino. There is no guarantee
28 * the circuit will work if the power supply is ON
29 * while you upload the wrong sketch to the Arduino.
30 *
31 * #####
32 *
33 *
34 * Arduino code to generate a signal with the AD9833 board.
35 * This program uses the serial interface of an Arduino NANO to receive data from a
36 * MATLAB application. This program receives a command from the MATLAB application
37 * specifying frequency, phase and waveform type of the wave we want to generate.
38 *
39 * THIS PROGRAM WORKS PERFECTLY IN ARDUINO SERIAL WINDOW
40 *
41 * AD9833 PINOUT:
42 *
43 * AD9833 VCC      -> Arduino 5V
44 * AD9833 DGND     -> No Connection
45 * AD9833 SDATA    -> Arduino MOSI Pin 11
46 * AD9833 SCLK     -> Arduino SCK Pin 13
47 * AD9833 FSYNC    -> User defined - For this script, Arduino Pin 10
48 * AD9833 AGND     -> Arduino GND
49 * AD9833 OUT      -> OUT Singal. To desired circuit
50 *
51 */
52
```

```

53 #include <AD9833.h> /*This library must be included in Arduino IDE for the program to
    work*/
54
55 #define FNC_PIN 10 /*FSYNC pin*/
56
57 /*Declare a signal generator*/
58 AD9833 signalGenerator(FNC_PIN);
59
60 /*other variables*/
61 float freq;
62 float phase;
63 int wave;
64 String serialData;
65 int i, j;
66 int indexCommas[2];
67
68 void setup() {
69     /*Initializing the signal generator. */
70     signalGenerator.Begin();
71     signalGenerator.ApplySignal(SINE_WAVE, REG0, 1000); /*set to a 1000Hz sine wave at
        program startup*/
72     signalGenerator.EnableOutput(true); /*Output is enabled*/
73
74     Serial.begin(9600);
75     /*Serial.print("\nMENU\n\nCOMMAND FORMAT: frequency,phase,waveformType\n\nWaveform type
        is:\n1.Sine\n2.Square\n3.Triangle\n4.Half-square\n\n");*/
76     Serial.print("\nMENU\n\nCOMANDO TIPO: frecuencia,fase,tipoOnda\n\nTipo de onda es:\n1.
        Seno\n2.Cuadrada\n3.Triangular\n4.Media Cuadrada\n\n");
77 }
78
79 void loop() {
80
81     if(Serial.available()){
82         /*Serial.flush(); //clear input buffer*/
83         /*serialData = Serial.readStringUntil('\r'); //Only for Arduino serial. When working
            with MATLAB:*/
84         serialData = Serial.readString();
85         /*Serial.println(data); //check (only in arduino IDE)*/
86         for(i = 0; i<serialData.length(); i++){
87             for(j = 0; j<2; j++){
88                 if(serialData[i] == ','){ /* A comma is read */
89                     indexCommas[j] = i;
90                 } /*ends IF*/
91             } /*ends FOR j*/
92         } /*ends FOR i*/
93
94         /* Variable asigation after reading serial*/
95         freq = serialData.substring(0, indexCommas[0]).toFloat();
96         phase = serialData.substring(indexCommas[0]+1, indexCommas[1]).toFloat();
97         wave = serialData.substring(indexCommas[1]+1, serialData.length()).toInt();
98
99         /*only for debugging in Arduino*/
100        Serial.println(freq);
101        Serial.println(phase);
102        Serial.println(wave);
103
104        /*setting phase*/
105        signalGenerator.SetPhase(REG0, phase);
106
107        /*applying signal to AD9833 depending on waveform type read from serial command*/
108        switch(wave){
109            case 1:
110                signalGenerator.ApplySignal(SINE_WAVE,REG0,freq);
111                break;
112
113            case 2:
114                signalGenerator.ApplySignal(SQUARE_WAVE,REG0,freq);
115                break;
116
117            case 3:
118                signalGenerator.ApplySignal(TRIANGLE_WAVE,REG0,freq);
119                break;
120
121            case 4:
122                signalGenerator.ApplySignal(HALF_SQUARE_WAVE,REG0,freq);

```

```
123     break;
124
125     default:
126     break;
127   }
128 }
129
130 }
```

Listing 1.1. Código de programación del AD9833

1.2. Código de programación potenciómetros y switches

```

1 /* setCurrent5Potentiometers
2  * @author Paula de Vega Botas
3  * July 25, 2019
4  *
5  * ##### IMPORTANT INFORMATION #####
6  *
7  * This program must be loaded to the Arduino NANO
8  * that is on the right side of the circuit board.
9  * If you're lost with this indication, look for the
10 * GND input of the whole circuit board. That's the
11 * side you're looking for.
12 *
13 * When working in the computer at IMDEA Arduino IDE
14 * recognised this Arduino NANO board in Port COM9,
15 * but it depends on which computer this program is
16 * going to be compiled and opened, so pay attention
17 * at which port recognises the Arduino NANO board when
18 * using another computer. Just keep in mind there are
19 * two Arduino NANO boards in this circuit and it is
20 * very important that each NANO gets the right sketch
21 * loaded. Otherwise the system will not work.
22 *
23 * #####
24 *
25 * This program does the following:
26 * - receives from serial interface a command sent by MATLAB. This command has the
   following format:
27 *   cmd = 'resistance1,resistance2,resistance3,resistance4,resistance5'
28 * - Adjust the values of the resistance of each potentiometer with the command
   received
29 * - Calculates how many switches to close depending on how many resistances have value
   0
30 * - Closes mentioned switches
31 *
32 * Every mapping operation between current and resistance is performed in MATLAB
33 *
34 *
35 * Wiring:
36 *
37 * MCP41100 #1 Pin 1 (CS) => Arduino Pin 10 (see note below)
38 * MCP41100 #1 Pin 2 (SCK) => Arduino Pin SCK (13 on Uno/Pro-Mini etc)
39 * MCP41100 #1 Pin 3 (SI) => Arduino Pin MOSI (11 on Uno/Pro-Mini etc)
40 * MCP41100 #1 Pin 4 (Vss) => Gnd
41 * MCP41100 #1 Pin 5 (Potentiometer B) => MCP41100 Pin 6 (Potentiometer Wiper)
42 * MCP41100 #1 Pin 6 (Potentiometer Wiper) => Multimeter Common Socket (Multimeter in
   resistance mode)
43 * MCP41100 #1 Pin 7 (Potentiometer A) => Multimeter Ohm's Socket (Multimeter in
   resistance mode)
44 * MCP41100 #1 Pin 8 (Vdd) => 5v
45 *
46 *
47 * MCP41100 #2 Pin 1 (CS) => Arduino Pin 9 (see note below)
48 * MCP41100 #2 Pin 2 (SCK) => Arduino Pin SCK (13 on Uno/Pro-Mini etc)
49 * MCP41100 #2 Pin 3 (SI) => Arduino Pin MOSI (11 on Uno/Pro-Mini etc)
50 * MCP41100 #2 Pin 4 (Vss) => Gnd
51 * MCP41100 #2 Pin 5 (Potentiometer B) => MCP41100 Pin 6 (Potentiometer Wiper)
52 * MCP41100 #2 Pin 6 (Potentiometer Wiper) => Multimeter Common Socket (Multimeter in
   resistance mode)
53 * MCP41100 #2 Pin 7 (Potentiometer A) => Multimeter Ohm's Socket (Multimeter in
   resistance mode)
54 * MCP41100 #2 Pin 8 (Vdd) => 5v
55 *
56 *
57 * MCP41100 #3 Pin 1 (CS) => Arduino Pin 6 (see note below)
58 * MCP41100 #3 Pin 2 (SCK) => Arduino Pin SCK (13 on Uno/Pro-Mini etc)
59 * MCP41100 #3 Pin 3 (SI) => Arduino Pin MOSI (11 on Uno/Pro-Mini etc)
60 * MCP41100 #3 Pin 4 (Vss) => Gnd
61 * MCP41100 #3 Pin 5 (Potentiometer B) => MCP41100 Pin 6 (Potentiometer Wiper)
62 * MCP41100 #3 Pin 6 (Potentiometer Wiper) => Multimeter Common Socket (Multimeter in
   resistance mode)
63 * MCP41100 #3 Pin 7 (Potentiometer A) => Multimeter Ohm's Socket (Multimeter in
   resistance mode)

```

```

64 * MCP41100 #3 Pin 8 (Vdd) => 5v
65 *
66 *
67 * MCP41100 #4 Pin 1 (CS) => Arduino Pin 4 (see note below)
68 * MCP41100 #4 Pin 2 (SCK) => Arduino Pin SCK (13 on Uno/Pro-Mini etc)
69 * MCP41100 #4 Pin 3 (SI) => Arduino Pin MOSI (11 on Uno/Pro-Mini etc)
70 * MCP41100 #4 Pin 4 (Vss) => Gnd
71 * MCP41100 #4 Pin 5 (Potentiometer B) => MCP41100 Pin 6 (Potentiometer Wiper)
72 * MCP41100 #4 Pin 6 (Potentiometer Wiper) => Multimeter Common Socket (Multimeter in
    resistance mode)
73 * MCP41100 #4 Pin 7 (Potentiometer A) => Multimeter Ohm's Socket (Multimeter in
    resistance mode)
74 * MCP41100 #4 Pin 8 (Vdd) => 5v
75 *
76 *
77 * MCP41100 #5 Pin 1 (CS) => Arduino Pin 5 (see note below)
78 * MCP41100 #5 Pin 2 (SCK) => Arduino Pin SCK (13 on Uno/Pro-Mini etc)
79 * MCP41100 #5 Pin 3 (SI) => Arduino Pin MOSI (11 on Uno/Pro-Mini etc)
80 * MCP41100 #5 Pin 4 (Vss) => Gnd
81 * MCP41100 #5 Pin 5 (Potentiometer B) => MCP41100 Pin 6 (Potentiometer Wiper)
82 * MCP41100 #5 Pin 6 (Potentiometer Wiper) => Multimeter Common Socket (Multimeter in
    resistance mode)
83 * MCP41100 #5 Pin 7 (Potentiometer A) => Multimeter Ohm's Socket (Multimeter in
    resistance mode)
84 * MCP41100 #5 Pin 8 (Vdd) => 5v
85 */
86
87 #include <Potenciometro.h> /* This library Potenciometro.h must be included in the
    Arduino IDE for the
88 code to work. */
89
90 Potenciometro MyPot1;
91 Potenciometro MyPot2;
92 Potenciometro MyPot3;
93 Potenciometro MyPot4;
94 Potenciometro MyPot5;
95
96 /* Which pin is connected to CS */
97 const uint8_t CS_PIN_POT1 = 10;
98 const uint8_t CS_PIN_POT2 = 9;
99 const uint8_t CS_PIN_POT3 = 6;
100 const uint8_t CS_PIN_POT4 = 4;
101 const uint8_t CS_PIN_POT5 = 5;
102
103 const float MAX_RES = 100000; /*Max resistance of the potentiometer: 100kOhm*/
104 const float WIPER_RES = 125; /*Wiper resistance of the potentiometer: around 125Ohm*/
105
106 void setup() {
107
108     /*pinout for the switch*/
109     pinMode(A0, OUTPUT);
110     pinMode(A1, OUTPUT);
111     pinMode(A2, OUTPUT);
112     pinMode(A3, OUTPUT);
113
114     /*Initializing the potentiometers*/
115     MyPot1.begin(CS_PIN_POT1);
116     MyPot2.begin(CS_PIN_POT2);
117     MyPot3.begin(CS_PIN_POT3);
118     MyPot4.begin(CS_PIN_POT4);
119     MyPot5.begin(CS_PIN_POT5);
120
121     Serial.begin(9600);
122     Serial.print("Introduce valor de la resistencia deseada de los cinco potenciómetros, \
        nseparados por una coma (SIN ESPACIOS) \nEJEMPLO: Introduce valor de las
        resistencias: 4500,6000,8600,3200,12369\n\n");
123     /*Serial.print("Introduce value of the desired resistance of all 5 potentiometers, \
        nseparated by a comma and NO SPACES\nExample: Introduce value of resistances:
        4500,6000,8600,3200,12369\n\n");*/
124 }
125
126 void loop() {
127
128     String res_pot1; /*resistance potentiometer 1*/
129     String res_pot2; /*resistance potentiometer 2*/

```

```

130 String res_pot3; /*resistance potentiometer 3*/
131 String res_pot4; /*resistance potentiometer 4*/
132 String res_pot5; /*resistance potentiometer 5*/
133
134 String command; /*command read from serial interface*/
135 int indexCommas[4]; /*to save the indexes where a comma appears in the "res" string
    read from serial port*/
136
137 if(Serial.available()>0)
138 {
139     Serial.print("\nIntroduce valor de las resistencias: ");
140     //Serial.print("\nIntroduce value of resistances: ");
141     Serial.flush(); //clear data input buffer
142     /*command = Serial.readStringUntil('\r'); //reads the string with the command with
    the value of both resistances.
143     //Works both for MATLAB and Serial interface of Arduino IDE, but I prefer to use
    Serial.readString() when
144     //the command is sent from MATLAB.*/
145     command = Serial.readString();
146     Serial.println(command);
147
148     /*Some control variables*/
149     int i,j;
150     j=0;
151
152     for(i = 0; i < command.length(); i++){
153         if(command[i] == ','){
154             /*a Comma is read*/
155             indexCommas[j] = i; /*I save the index*/
156             j++;
157         }
158     }
159
160     /*Separate the command read from the serial interface into all 5 values of
    resistance
161     * for all 5 potentiometers
162     */
163     res_pot1 = command.substring(0,indexCommas[0]);
164     res_pot2 = command.substring(indexCommas[0]+1, indexCommas[1]);
165     res_pot3 = command.substring(indexCommas[1]+1, indexCommas[2]);
166     res_pot4 = command.substring(indexCommas[2]+1, indexCommas[3]);
167     res_pot5 = command.substring(indexCommas[3]+1, command.length());
168
169     Serial.println("\nPotenciometro 1: ");
170     Serial.print(res_pot1.toFloat());
171     Serial.println("\nPotenciometro 2: ");
172     Serial.print(res_pot2.toFloat());
173     Serial.println("\nPotenciometro 3: ");
174     Serial.print(res_pot3.toFloat());
175     Serial.println("\nPotenciometro 4: ");
176     Serial.print(res_pot4.toFloat());
177     Serial.println("\nPotenciometro 5: ");
178     Serial.print(res_pot5.toFloat());
179
180     /*Set the value of the resistance read from serial to each potentiometer*/
181     MyPot1.setWiper(((res_pot1.toFloat()-WIPE_RES)/MAX_RES) * 255);
182     MyPot2.setWiper(((res_pot2.toFloat()-WIPE_RES)/MAX_RES) * 255);
183     MyPot3.setWiper(((res_pot3.toFloat()-WIPE_RES)/MAX_RES) * 255);
184     MyPot4.setWiper(((res_pot4.toFloat()-WIPE_RES)/MAX_RES) * 255);
185     MyPot5.setWiper(((res_pot5.toFloat()-WIPE_RES)/MAX_RES) * 255);
186
187     /*Select the switches to close*/
188     int nSwitches = 0;
189
190     if(res_pot2.toInt() != 0){
191         nSwitches++;
192     }
193     if(res_pot3.toInt() != 0){
194         nSwitches++;
195     }
196     if(res_pot4.toInt() != 0){
197         nSwitches++;
198     }
199     if(res_pot5.toInt() != 0){
200         nSwitches++;

```

```
201     }
202
203     cerrarSwitches(nSwitches);
204 }
205 }
206
207
208 void cerrarSwitches(int numSwitches){
209     int i,j;
210     /* Swich inputs are connected to A0, A1, A2 and A3
211      * but are established as Digital Pins 14, 15, 16 and 17
212      */
213
214     for(j = 0; j<4; j++){
215         digitalWrite(j+14, HIGH); /*First all switches are open*/
216     }
217
218     for(i = 0; i<numSwitches; i++){
219         digitalWrite(i+14, LOW);
220     }
221 }
```

Listing 1.2. Código que controla los potenciómetros y los switches

Chapter 2

Matlab

2.1. Función de mapeo corriente a resistencia

```
1 function [ resistance ] = mapCurrentToResistance( current )
2 % MapCurrentToResistance function
3 % This function maps the values of current that the user asks for, taken
4 % from the user interface and each current value is assigned to a
5 % corresponding resistance value. This resistance value generates the
6 % necessary gain for the circuit to supply the desired current.
7 % Input arguments are:
8 %     - user input current
9 % Output arguments are:
10 %     - resistance assigned to user input current
11
12 % Esta funcion mapea los valores de corriente que tome de la interfaz grafica
13 % que pida el usuario y los asigna con el correspondiente valor de resistencia
14 % que genera la ganancia necesaria para que el circuito suministre dicha corriente
15 % argumentos de entrada:
16 %     - corriente que pide el usuario
17 % argumentos de salida:
18 %     - resistencia asignada a dicha corriente
19
20     currentMatrix = importdata('currentMatrix.mat');
21     randomCurrentValue = current;
22     resistanceForRandomCurrent1 = 0;
23     resistanceForRandomCurrent2 = 0;
24     epsilon = 0.3;
25     littleEpsilon = 0.05;
26     roundedValue = round(randomCurrentValue,1); %rounds to closes decimal
27
28     for i = 1:length(currentMatrix(:,1)) %for 1:65
29         %length(currentMatrix(1,:)) = 2
30         %first assert if value is closer
31         if((round(currentMatrix(i,2),2)-littleEpsilon) <= randomCurrentValue)
32             if((randomCurrentValue <= round(currentMatrix(i,2),2)+littleEpsilon))
33                 resistanceForRandomCurrent1 = currentMatrix(i,1);
34             end
35         end
36
37         %do a second test if last one was not correct or if there is very
38         %little range in the first comprobation, as the assignation is
39         %non-linear
40         if((round(currentMatrix(i,2),1)-epsilon <= roundedValue))
41             if((roundedValue <= round(currentMatrix(i,2),1)+epsilon))
42                 resistanceForRandomCurrent2 = currentMatrix(i,1);
43             end
44         end
45     end
46
47     %check whether current input is not 0. Also, checks if first comprobation was
48     %accurate and there was a resistance value assigned. If the resistance
49     %value for the first loop is zero it means the comprobation was too
50     %accurate for the value, so it goes on to the next for loop and assigns
51     %a value for resistance in that second for loop.
52     if(randomCurrentValue ~= 0)
```

```
53     if((resistanceForRandomCurrent1 < resistanceForRandomCurrent2) &&  
        resistanceForRandomCurrent1 ~= 0)  
54         resistance = resistanceForRandomCurrent1;  
55     else%(resistanceForRandomCurrent1 == 0)  
56         resistance = resistanceForRandomCurrent2;  
57     end  
58 else  
59     resistance = 0;  
60 end  
61  
62 end
```

Codigo/mapCurrentToResistance.m

2.2. Función de generación de comando a enviar al Arduino

```

1 function [command] = generateCommand ( corrienteInput )
2 % funtion generateCommand
3
4 maxCorrientePorCanal = 22; %mA
5 nCanalesATope = floor(corrienteInput/maxCorrientePorCanal);
6 numeroCanales = 1; %At least there will be one channel always working
7
8 if(corrienteInput - (nCanalesATope*maxCorrientePorCanal) == 0)
9     %All channels are working at their maximum current possible
10     numeroCanales = nCanalesATope;
11 else
12     %We add one more channel to work at its maximum current possible
13     numeroCanales = nCanalesATope +1;
14 end
15
16 %We initialize all potentiometers to 10kOhms just in case
17 res_pot1 = 10000;
18 res_pot2 = 10000;
19 res_pot3 = 10000;
20 res_pot4 = 10000;
21 res_pot5 = 10000;
22
23
24 %We use previously defined mapCurrentToResistance function to assign the
25 %current each potentiometer is going to supply. This function returns the
26 %value of the resistance each potentiometer must take in order to supply
27 %the current of the input argument.
28
29 switch numeroCanales
30     case 1
31         res_pot1 = mapCurrentToResistance(corrienteInput);
32         res_pot2 = mapCurrentToResistance(0);
33         res_pot3 = mapCurrentToResistance(0);
34         res_pot4 = mapCurrentToResistance(0);
35         res_pot5 = mapCurrentToResistance(0);
36     case 2
37         res_pot1 = mapCurrentToResistance(maxCorrientePorCanal);
38         res_pot2 = mapCurrentToResistance(corrienteInput - maxCorrientePorCanal);
39         res_pot3 = mapCurrentToResistance(0);
40         res_pot4 = mapCurrentToResistance(0);
41         res_pot5 = mapCurrentToResistance(0);
42     case 3
43         res_pot1 = mapCurrentToResistance(maxCorrientePorCanal);
44         res_pot2 = mapCurrentToResistance(maxCorrientePorCanal);
45         res_pot3 = mapCurrentToResistance(corrienteInput - 2*maxCorrientePorCanal);
46         res_pot4 = mapCurrentToResistance(0);
47         res_pot5 = mapCurrentToResistance(0);
48     case 4
49         res_pot1 = mapCurrentToResistance(maxCorrientePorCanal);
50         res_pot2 = mapCurrentToResistance(maxCorrientePorCanal);
51         res_pot3 = mapCurrentToResistance(maxCorrientePorCanal);
52         res_pot4 = mapCurrentToResistance(corrienteInput - 3*maxCorrientePorCanal);
53         res_pot5 = mapCurrentToResistance(0);
54     case 5
55         res_pot1 = mapCurrentToResistance(maxCorrientePorCanal);
56         res_pot2 = mapCurrentToResistance(maxCorrientePorCanal);
57         res_pot3 = mapCurrentToResistance(maxCorrientePorCanal);
58         res_pot4 = mapCurrentToResistance(maxCorrientePorCanal);
59         res_pot5 = mapCurrentToResistance(corrienteInput - 4*maxCorrientePorCanal);
60 end
61
62
63 %This is the command that will be sent to the Arduino NANO. It controls the
64 %resistance each potentiometer will take and depending on how many
65 %potentiometers have its value different to zero, how many swtches will be
66 %closed
67 command = [num2str(res_pot1), ',', num2str(res_pot2), ',', num2str(res_pot3), ',',
68             num2str(res_pot4), ',', num2str(res_pot5)];

```

69 end

Codigo/generateCommand.m

2.3. Código de la aplicación de interfaz gráfica

```

1 classdef AD9833 < matlab.apps.AppBase
2
3     % Properties that correspond to app components
4     properties (Access = public)
5         UIFigure                matlab.ui.Figure
6         LabelNumericEditField    matlab.ui.control.Label
7         Frequency                matlab.ui.control.NumericEditField
8         LabelNumericEditField2    matlab.ui.control.Label
9         Phase                    matlab.ui.control.NumericEditField
10        StartApp                 matlab.ui.control.Button
11        WaveformType             matlab.ui.container.ButtonGroup
12        Sine                     matlab.ui.control.RadioButton
13        Square                   matlab.ui.control.RadioButton
14        Triangle                 matlab.ui.control.RadioButton
15        HalfSquare               matlab.ui.control.RadioButton
16        DesiredCurrent5mAto100mALabel matlab.ui.control.Label
17        DesiredCurrent            matlab.ui.control.NumericEditField
18    end
19
20    methods (Access = private)
21
22        % Code that executes after component creation
23        function startupFcn(app)
24            clc, close all, clear all;
25        end
26
27        % Value changed function: Frequency
28        function ChangeFrequency(app, event)
29            global freq;
30            % get the value of the frequency from the text box from the UI
31            freq = app.Frequency.Value;
32            display(freq);
33        end
34
35        % Value changed function: Phase
36        function ChangePhase(app, event)
37            global phase;
38            % get the value of the phase from the text box from the UI
39            phase = app.Phase.Value;
40            display(phase);
41        end
42
43        % Selection changed function: WaveformType
44        function ChoseWaveform(app, event)
45            global waveform;
46            selectedButton = app.WaveformType.SelectedObject;
47            if(strcmp(selectedButton.Text, 'Sine'))
48                % We send sine wave to serial interface
49                waveform = 1;
50            end
51            if(strcmp(selectedButton.Text, 'Square'))
52                % We send square wave to serial interface
53                waveform = 2;
54            end
55            if(strcmp(selectedButton.Text, 'Triangle'))
56                % We send triangle wave to serial interface
57                waveform = 3;
58            end
59            if(strcmp(selectedButton.Text, 'Half Square'))
60                % We send half square wave to serial interface
61                waveform = 4;
62            end
63        end
64
65        % Button pushed function: StartApp
66        function Start(app, event)
67            % In function Start we will be sending all variable values to the serial
68            interface for both Arduino NANO boards
69
70            %Variables declared as global because they are being used in other functions
71            (i.e. their value is taken

```

```

70     %from the user interface in another function
71     global freq;
72     global phase;
73     global waveform;
74     global serial_connection1;
75     global serial_connection2;
76     global resistance_command;
77
78     % We declare the serial connections for both Arduino NANO boards
79     % The value of the COM ports should be changed when working with another computer
80     serial_connection1 = serial('COM5', 'BAUD', 9600);
81     serial_connection2 = serial('COM9', 'BAUD', 9600);
82
83     command = [num2str(freq), ',', num2str(phase), ',', num2str(waveform)];
84     display(command); %to debug
85
86     %Open serial connection 1, wait for it to open, write the command to the serial
interface
87     %wait for the command to be sent and then close the serial connection
88     % It is a good practice to open and close the serial connection for each operation
89     %we perform. Otherwise leaving the port open may cause some problems and
inconveniences
90     %that will end up making the user close MATLAB completely and opening it again
91     fopen(serial_connection1);
92     pause(2);
93     fprintf(serial_connection1, command);
94     pause(2);
95     fclose(serial_connection1);
96
97     %Same for serial connection 2
98     fopen(serial_connection2);
99     pause(2);
100    fprintf(serial_connection2, resistance_command);
101    pause(2);
102    fclose(serial_connection2);
103
104    end
105
106    % Value changed function: DesiredCurrent
107    function getDesiredCurrent(app, event)
108        global current;
109        global resistance_command;
110        current = app.DesiredCurrent.Value;
111        % We use generateCommand function previously defined
112        resistance_command = generateCommand(current);
113        display(resistance_command); %To debug
114
115    end
116    end
117
118    % App initialization and construction
119    methods (Access = private)
120
121        % Create UIFigure and components
122        function createComponents(app)
123
124            % Create UIFigure
125            app.UIFigure = uifigure;
126            app.UIFigure.Position = [101 101 512 336];
127            app.UIFigure.Name = 'UI Figure';
128
129            % Create LabelNumericEditField
130            app.LabelNumericEditField = uilabel(app.UIFigure);
131            app.LabelNumericEditField.HorizontalAlignment = 'right';
132            app.LabelNumericEditField.VerticalAlignment = 'top';
133            app.LabelNumericEditField.Position = [278 237 57 15];
134            app.LabelNumericEditField.Text = 'Frequency';
135
136            % Create Frequency
137            app.Frequency = uieditfield(app.UIFigure, 'numeric');
138            app.Frequency.Limits = [0 12000000000];
139            app.Frequency.ValueChangedFcn = createCallbackFcn(app, @ChangeFrequency, true);
140            app.Frequency.Position = [350 233 78 22];
141
142            % Create LabelNumericEditField2

```

```

143     app.LabelNumericEditField2 = uilabel(app.UIFigure);
144     app.LabelNumericEditField2.HorizontalAlignment = 'right';
145     app.LabelNumericEditField2.VerticalAlignment = 'top';
146     app.LabelNumericEditField2.Position = [278 193 35 15];
147     app.LabelNumericEditField2.Text = 'Phase';
148
149     % Create Phase
150     app.Phase = uieditfield(app.UIFigure, 'numeric');
151     app.Phase.ValueChangedFcn = createCallbackFcn(app, @ChangePhase, true);
152     app.Phase.Position = [350 191 78 20];
153
154     % Create StartApp
155     app.StartApp = uibutton(app.UIFigure, 'push');
156     app.StartApp.ButtonPushedFcn = createCallbackFcn(app, @Start, true);
157     app.StartApp.BackgroundColor = [0.4863 0.8667 0.4863];
158     app.StartApp.Position = [210 47 100 22];
159     app.StartApp.Text = 'Start';
160
161     % Create WaveformType
162     app.WaveformType = uibuttongroup(app.UIFigure);
163     app.WaveformType.SelectionChangedFcn = createCallbackFcn(app, @ChoseWaveform,
true);
164     app.WaveformType.Title = 'Waveform Type';
165     app.WaveformType.FontName = 'MS Sans Serif';
166     app.WaveformType.Position = [82 164 121 117];
167
168     % Create Sine
169     app.Sine = uiradiobutton(app.WaveformType);
170     app.Sine.Text = 'Sine';
171     app.Sine.Position = [11 71 43.03125 16];
172     app.Sine.Value = true;
173
174     % Create Square
175     app.Square = uiradiobutton(app.WaveformType);
176     app.Square.Text = 'Square';
177     app.Square.Position = [11 49 57.703125 16];
178
179     % Create Triangle
180     app.Triangle = uiradiobutton(app.WaveformType);
181     app.Triangle.Text = 'Triangle';
182     app.Triangle.Position = [11 26 62.359375 16];
183
184     % Create HalfSquare
185     app.HalfSquare = uiradiobutton(app.WaveformType);
186     app.HalfSquare.Text = 'Half Square';
187     app.HalfSquare.Position = [11 3 82.375 16];
188
189     % Create DesiredCurrent5mAto100mALabel
190     app.DesiredCurrent5mAto100mALabel = uilabel(app.UIFigure);
191     app.DesiredCurrent5mAto100mALabel.HorizontalAlignment = 'right';
192     app.DesiredCurrent5mAto100mALabel.Position = [108 109 181 22];
193     app.DesiredCurrent5mAto100mALabel.Text = 'Desired Current (5mA to 100mA)';
194
195     % Create DesiredCurrent
196     app.DesiredCurrent = uieditfield(app.UIFigure, 'numeric');
197     app.DesiredCurrent.ValueChangedFcn = createCallbackFcn(app, @getDesiredCurrent,
true);
198     app.DesiredCurrent.Position = [304 109 100 22];
199     end
200 end
201
202 methods (Access = public)
203
204     % Construct app
205     function app = AD9833
206
207         % Create and configure components
208         createComponents(app)
209
210         % Register the app with App Designer
211         registerApp(app, app.UIFigure)
212
213         % Execute the startup function
214         runStartupFcn(app, @startupFcn)
215

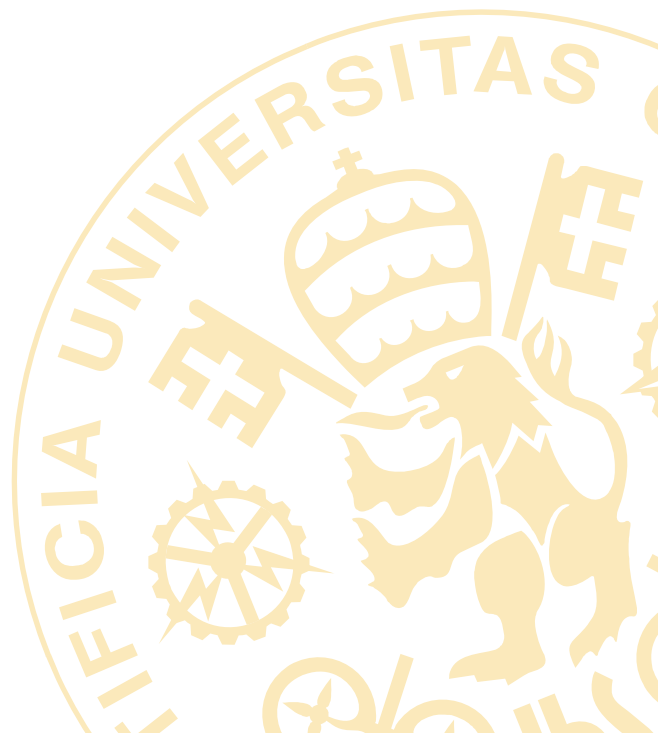
```

```
216         if nargin == 0
217             clear app
218         end
219     end
220
221     % Code that executes before app deletion
222     function delete(app)
223
224         % Delete UIFigure when app is deleted
225         delete(app.UIFigure)
226     end
227 end
228 end
```

Codigo/aplicacionfinalenCodigoMatlab.m

PART III

ESTUDIO ECONÓMICO



Chapter 1

Estudio económico

EN esta sección de la memoria se va a proceder a hacer un breve estudio económico del proyecto.

1.1. Componentes electrónicos

En primer lugar, se enunciarán los componentes electrónicos y los correspondientes precios y unidades. En la tabla que figura a continuación se hace lo enunciado. Para más referencia acerca de los componentes buscar en el capítulo 1, en el apartado de **Recursos** qué es cada componente.

Componente	Nº unidades	Precio/ud (€)	Total
Arduino NANO	2	20.00	40.00
OPA445A	15	10.00	150.00
TL072ACP	10	0.51	5.10
ADG444	1	3.00	3.00
MCP41100	15	1.25	18.75
AD9833	1	4.00	4.00
MJE3055	5	1.95	9.75
ARCOL 250 Ω	1	6.62	6.62
AGILENT 34410	1	300.00	300.00
TOTAL	-	-	537.22

Table 3. Tabla de precios de componentes

1.2. Licencias de software

En este apartado se enuncian las licencias de software que ha sido necesario pagar para poder implementar este sistema.

Software	Precio (€)
LabVIEW Software	1500.00
MATLAB Software	1000.00
TOTAL	2500.00

Table 4. Tabla de precios de software

Nótese que el software de Arduino no ha sido enunciado ya que se trata de una empresa de código abierto y distribuyen su IDE de manera gratuita.

1.3. Suma total

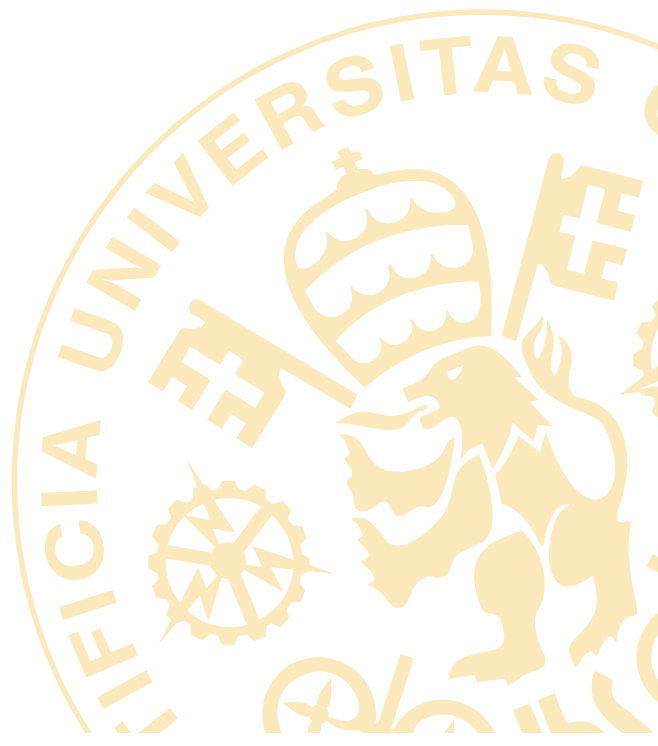
A continuación se calcula la suma total de los componentes y de las licencias de software para calcular el precio final del proyecto.

Item	Precio (€)
Componentes electrónicos	537.22
Licencias de Software	2500.00
TOTAL	3037.22

Table 5. Tabla de suma total de costes

PART IV

MANUAL DE USUARIO



Conexiones

El sistema está encapsulado en una caja con 5 entradas y 2 salidas.

Alimentación del circuito

- La alimentación del circuito es sencilla. Hay que usar la fuente de voltaje a $\pm 25V$ y conectada en serie. Hay que conectar las tierras a nodo común.
- La tensión positiva de +25V se conecta a la entrada de Vcc, +25V, del sistema.
- La tensión negativa de -25V se conecta a la entrada de Vee, -25V, del sistema.
- El nodo de tierra, GND se conecta a la entrada de GND del sistema.

Es muy importante conectar la alimentación bien ya que si no el sistema no funcionará y se correrá el riesgo de cortocircuitos o de estropear algún componente del sistema.

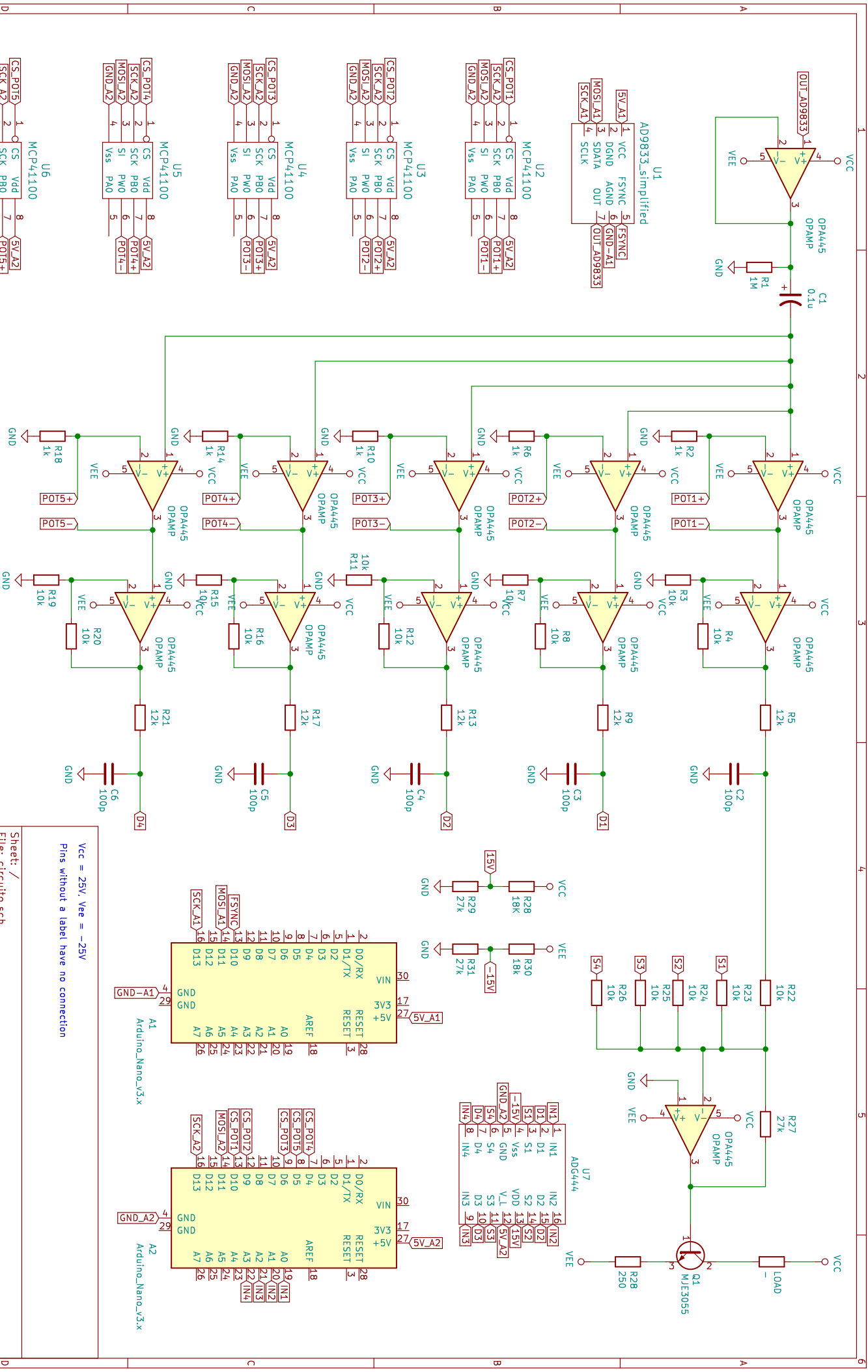
Conexión de los Arduinos

- Arduino NANO 1 tiene que tener cargado el programa de Arduino AD9833.
- Arduino NANO 2 tiene que tener cargado el programa de Arduino setCurrent5Potentiometers.

Es importante cargar los programas correspondientes a cada Arduino ya que si no se garantiza el funcionamiento correcto del circuito. **No cargar los programas mientras la alimentación esté conectada al circuito.**

Salidas del sistema

El sistema tiene dos salidas, etiquetadas como **CÁTODO** y **ÁNODO**. El circuito queda abierto hasta que las salidas queden conectadas a algún dispositivo (o incluso cortocircuitarlas entre sí). Aunque las etiquetas son de cátodo ánodo se puede conectar cualquier dispositivo: un cortocircuito, una resistencia, un diodo LED o un cristal LEC. Para estos dos últimos es importante colocar la polaridad correctamente para que el circuito funcione.



Vcc = 25V, Vee = -25V
Pins without a label have no connection

Sheet: /
File: circuito.sch

Title: FINAL CIRCUIT

Size: A4 Date: August 8th, 2019

KiCad E.D.A. kicad (5.1.2)-2

Rev: 2.0

Id: 1/1