



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño e implementación de un sintetizador de audio modular basado en síntesis substractiva

TRABAJO DE FIN DE GRADO

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Gutiérrez-Ravé Olmos, Jorge

Tutor: Coll Arnau, Salvador

Agradecimientos:

Quisiera agradecer a varias personas y gente de mi entorno la ayuda que me han dedicado para llevar a cabo este proyecto. Gracias a ellos he conseguido la fuerza, conocimiento, y ganas necesarias para afrontar este gran plan.

A mi familia, por creer en mí y apoyarme con mi proyecto sin dudar, desde la ayuda económica hasta el apoyo moral.

A Claudia Vanacloig, por prestarme su incalculable ayuda en el proyecto, sobre todo por ocuparse del diseño y la imagen, su personal visión del diseño ha dado vida a lo que es NANO hoy en día.

A Salvador Coll, por haber visto potencial en este proyecto y desde el día uno apoyarlo con su ayuda y conocimiento, agradezco mucho que me haya dedicado tanto tiempo para darle forma a este sistema de síntesis, sobre todo el apoyo que me ha dado su experiencia.

A Andrés Vallés, por ser una fuente de ayuda inagotable durante los años que he estado en la ETSID. Su ayuda en forma de ideas ha sido muy útil para dar forma a este proyecto y a muchos otros que había hecho antes.

A mis amigas, por apoyar el proyecto y mostrar su interés por él, ya que el apoyo de la gente de mi entorno ha sido muy importante para poder dedicarme a ello sin dudar. Estoy muy contento y agradecido con el apoyo recibido por parte de la gente que conozco.

A la gente anónima que sigue el proyecto en las redes sociales, animando a que nos esforcemos para continuar sacando nuevos productos. La gran acogida que ha tenido en redes sociales ha sido extraordinaria y nos anima a continuar con ello.

A la gente que suma y que te hacen crecer como persona, a la gente con la que aprendes y a la gente que cree en tí.

Índice

Agradecimientos	3
1. Definición del proyecto	7
1.1. Breve descripción del proyecto:	7
1.2. Objetivos y retos planteados:	7
2. Introducción a la síntesis de sonido	11
2.1. Análisis general:	11
2.2. Profundización en la síntesis sustractiva:	12
3. Análisis del mercado de sintetizadores modulares	17
3.1. Breve análisis de mercado:	17
3.2. Análisis del cliente:	18
3.3. Análisis de competencia:	19
3.4. Tecnología disponible:	20
3.5. Posibilidad de innovación en este mercado:	21
3.6. Explicación de la elección de este mercado:	21
3.7. Conclusiones:	22
4. Especificación de requisitos	23
4.1. Introducción:	23
4.2. Propósito:	23
4.3. Ámbito del Sistema:	23
4.4. Definición de los requisitos:	24
4.5. Definiciones, acrónimos y abreviaturas:	24
5. Descripción general	27
5.1. Funciones del producto:	27
5.2. Características de los usuarios:	28
5.3. Restricciones:	28
6. Requisitos específicos.	29
6.1. Requisitos de rendimiento:	29
6.2. Restricciones de diseño:	30
6.3. Atributos mecánicos del sistema:	32
7. Entorno de desarrollo.	35
7.1. Introducción:	35
7.2. Tecnologías utilizadas:	35
7.2.1. Hardware:	35
7.2.2. Software:	35

8. Diseño.	37
8.1. Introducción:	37
8.2. Entradas y salidas:	37
8.3. Software:	44
8.4. Hardware:	50
8.4.1. Diseño del esquema electrónico:	50
	57
8.4.2. Diseño de la PCB:	67
8.5. Componentes mecánicos:	69
8.5.1. Diseño de los paneles:	69
9. Implementación:	71
9.1. Prototipado en placa board:	71
9.2. Diseño del hardware:	73
9.3. Diseño del software:	75
9.4. Diseño del panel:	77
10. Validación del dispositivo:	79
10.1. Introducción	79
10.2. Testeo del hardware	79
10.3. Testeo de software	91
10.4. Testeo de elementos mecánicos	100
11. Resumen del trabajo desarrollado:	103
11.1. Trabajo previo	103
11.2. Sistema final	106
11.3. Marca NANO	107
11.4. Trabajo futuro	108
11.5. Conclusiones	109
11.5.1. Verificación de funcionamiento.	109
11.5.2. Conclusión personal.	111
12. Bibliografía	113

1. Definición del proyecto

1.1. Breve descripción del proyecto:

Nuestro propósito es el diseño e implementación de un sistema de síntesis sustractiva. Para lograr este objetivo, vamos a fabricar los módulos presentados anteriormente para contar con un sistema completo de síntesis.

No obstante, vamos a dar diferente enfoque a estos módulos, centrando la mayor parte de la atención en el oscilador de frecuencia de audio (**VCO**), ya que es el corazón de la síntesis sustractiva, y el (**LFO**), oscilador de baja frecuencia, complementario al VCO y que comparte similitud tanto en hardware como en software.

Los demás módulos servirán de complemento a estos, permitiendo modular las señales y cambiar sus atributos para extraer sonidos armónicos. Estos módulos se han diseñado, prototipado, y fabricado para poder tener un sistema tangible con el cual podemos interactuar y crear con él.

1.2. Objetivos y retos planteados:

El principal objetivo planteado para este proyecto es la creación de un sistema básico de síntesis sustractiva, esto se conseguirá por medio de diseño, test e implementación de un conjunto de módulos que permita configurar un sintetizador de audio funcional.

El principal reto del proyecto será el mejorar las funcionalidades existentes en módulos comerciales. En este proyecto, nos planteamos mejoras respecto a los módulos equivalentes a nuestros diseños (**VCO, LFO, VCA, ADSR y VCF**) Estas mejoras nos ayudarán a despuntar por encima de los demás módulos, generando interés sobre los nuestros.

Esta mejora de las características será determinada en el apartado de **Especificación de requisitos**. En conclusión, lo que buscamos a la hora del TFG no es igualar las características ya presentes en estos módulos, es mejorarlas.

1.3. Descripción de las partes del documento:

Este trabajo incluye varios documentos en los cuales se presenta toda la información necesaria para llevar a cabo el diseño e implementación de un sintetizador de audio modular basado en síntesis sustractiva. Los documentos son los siguientes:

- **Memoria:** Documento donde se encuentra la información teórica sobre el proyecto, desde la definición hasta el diseño y la validación de su funcionamiento.

- **Pliego de condiciones:** Documento en que se exponen las condiciones a las que deben sujetarse el contrato de la fabricación del dispositivo. Contiene información sobre requerimientos de materiales.
- **Presupuesto:** Documento donde se encuentran los presupuestos de cada módulo y el coste total del sistema de síntesis.
- **Planos:** Documento en el que se muestran los planos del sistema de síntesis, tanto mecánicos como eléctricos.
- **Código:** Documento en el que se muestra el código empleando en el desarrollo del proyecto, ya sea como librerías propias como código exclusivo de cada módulo.

En cuanto a la memoria, el documento está dividido en varias partes, cada una se encarga de explicar los diferentes procesos seguidos para la implementación y el diseño de este sistema de síntesis sustractiva:

1. **Definición del proyecto:** Breve definición del propósito del proyecto donde se comenta los propósitos y objetivos.
2. **Introducción a la síntesis de sonido:** Explicación de la síntesis de audio, necesaria para entender en qué se basa nuestro proyecto. Se profundiza en la síntesis sustractiva ya que es exactamente a lo que dedicamos este sistema, se enumeran los módulos principales y se explica su funcionamiento.
3. **Análisis del mercado de sintetizadores modulares:** Profundización en el mercado de los sintetizadores modulares, donde se hace una pequeña introducción para entrar en materia. Después de esta introducción se analizan a los posibles clientes y a la competencia. Posteriormente se habla sobre la tecnología disponible y la capacidad de innovación de este mercado, finaliza con unas conclusiones.
4. **Especificación de requisitos:** En este apartado se incluye una definición del sistema, se indica su propósito y ámbito. También se incluyen las definiciones y abreviaturas que hagan falta ser explicadas, y concluye con una descripción de las partes del documento.
5. **Descripción general:** Profundización en la definición del sistema de síntesis, ya que se presentan sus funciones, las características de los usuarios, y las restricciones presentes.

- 6. Requisitos específicos:** Evaluación de los requisitos que tiene el sistema, primero se analizan los requisitos de rendimiento, posteriormente los de diseño, y finaliza con los atributos mecánicos.
- 7. Entorno de desarrollo:** Pretende dar información acerca de las herramientas de software que se utilizan para desarrollar el sistema de síntesis. Se diferencia entre herramientas para diseño de hardware y software. Concluye con una breve explicación de cada entorno.
- 8. Diseño:** Apartado donde se muestra el funcionamiento de los diferentes dispositivos, primero el software y luego el hardware. Finalmente se muestran los paneles diseñados para cada módulo.
- 9. Implementación:** Se muestra el proceso de diseño seguido para elaborar el sistema de síntesis, desde el primer prototipo hasta tener ya módulos completamente funcionales.
- 10. Validación del dispositivo:** Apartado donde se comprueban las características y funcionalidades de los módulos creados. Se comprueba el hardware, software y los componentes mecánicos.
- 11. Resumen del trabajo desarrollado:** Se muestra el trabajo previo a la creación de este proyecto, también se habla de la creación de la marca NANO y acaba con una breve explicación del trabajo futuro.
- 12. Bibliografía:** Referencias bibliográficas incluidas en el documento, provenientes de libros y de páginas web.

2. Introducción a la síntesis de sonido

2.1. Análisis general:

La síntesis de sonido consiste en obtener ondas sonoras a partir de procesos no acústicos. Estas variaciones pueden ser de diferentes naturalezas:

- **Analógica:** Variaciones de voltajes que forman ondas sonoras, generado por complejos circuitos de naturaleza analógica [1].
- **Digital:** Producen sonido por medio de programas de circuitos integrados digitales [1].

Existen diferentes métodos de síntesis, entre ellos:

- **Síntesis aditiva:** Consiste en la superposición de ondas fundamentales para formar ondas complejas, un ejemplo de esto sería un órgano de tubos. Es muy frecuente utilizar ondas senoidales y sus propios armónicos para crear sonido [2].
- **Síntesis sustractiva:** Se da forma al sonido mediante la filtración y amplificación de una onda compleja. La señal pasa a través de diferentes etapas que modifican su contenido armónico, atenuando o reforzando determinadas áreas del espectro de la señal, también se varía su amplitud para conseguir tonos musicales [3].
- **Síntesis por modulación:** Esta modulación comprende los métodos en los que se modifica parámetros de las ondas en base a otra onda moduladora para generar señales con un espectro complejo. Esta modulación se puede dar de varias formas, siendo estas dos las más habituales:
 - **Modulación de amplitud (AM),** consiste en modificar la amplitud de la onda portadora en función de la onda moduladora, esto como resultado genera una señal de amplitud variable [4].
 - **Modulación de frecuencias (FM),** consiste modificar la frecuencia de una onda portadora en función de la forma la onda moduladora, este proceso genera una señal de frecuencia variable [5].
- **Síntesis por modelos físicos:** método de síntesis que produce sonido a partir de una simulación digital de un proceso físico [6].
- **Síntesis granular:** Este tipo de síntesis trata de formar sonido con unos elementos mínimos que son fragmentos de ondas, llamados 'granos'. Al repetir estos granos de forma rítmica se puede dar forma al sonido consiguiendo resultados agradables de escuchar [7].

Los métodos mencionados son los más habituales, pero no son los único. Este campo es muy amplio, y se avanza día a día.

2.2. Profundización en la síntesis sustractiva:

Fabricar un dispositivo de **síntesis sustractiva** es el propósito de este proyecto, por tanto, vamos a abordar detalladamente el funcionamiento de la misma [8].

La síntesis sustractiva, como su propio nombre indica, trata de sustraer a una onda principal diferentes parámetros para darle forma de sonido musical. Esta sustracción principalmente es de armónicos de la propia señal y de amplitud.

Estos sistemas se basan en diferentes etapas que dan forma a la señal, tiene múltiples posibilidades al permitir hacer un enrutado con cables jack, en la [Figura 1] se muestra un enrutado básico para producir sonido:

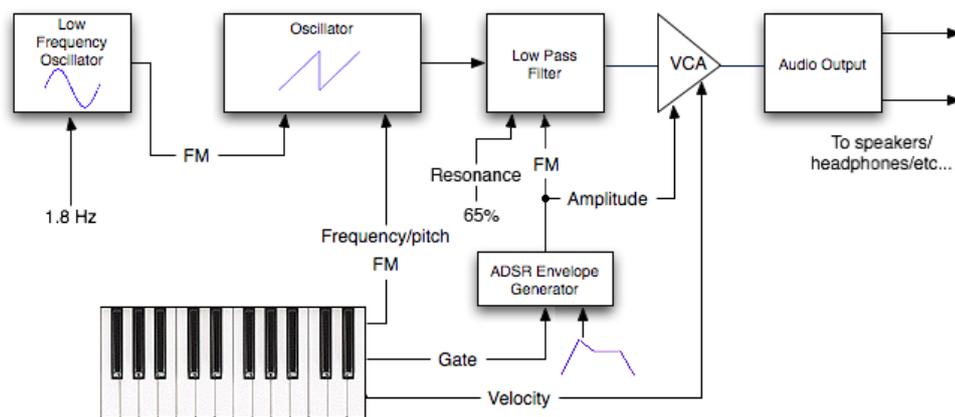


Figura 1 - Diagrama de bloques básico de nuestro sistema

En este diagrama se puede observar cómo se utilizan todos los módulos que engloba el proyecto para dar forma al sonido. En primer lugar, tenemos el **LFO** y el teclado que ajustan la frecuencia del **VCO**. A la salida del VCO, la señal de audio pasa por el **VCF**, controlado por el generador de **Envolvente**, que se encarga de abrir o cerrar el filtro dependiendo de si la tecla del teclado está presionada o no. Finalmente, la señal de audio pasa por el **VCA**, que se encarga de amplificar y dar una salida ya preparada para ser escuchada por altavoces o auriculares.

Siguiendo el diagrama de bloques, podemos concluir que los módulos encargados de generar estas funciones que nos van a permitir producir sonido, son los siguientes:

- **Osciladores de frecuencia de audio (VCO):** Es la fuente de sonido en la síntesis sustractiva. Generan la señal a la cual se le van a modificar parámetros como la amplitud o el contenido armónico para dar forma al sonido final.

Los osciladores generan señales básicas como senoidales, triangulares, cuadradas, y rampas, como las observadas en la [Figura 2].

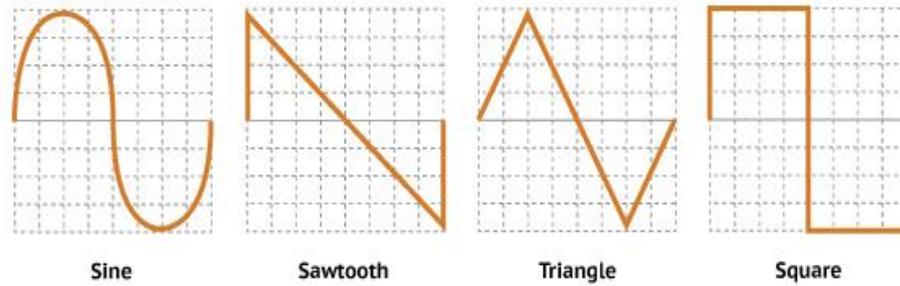


Figura 2 - Formas de onda más comunes

Estas señales son variables en frecuencia en el rango audible [20, 20.000]Hz, ajustándose al tono requerido.

Generalmente estos osciladores están controlados por voltaje, permitiendo ajustar la frecuencia desde entradas analógicas.

- **Osciladores de baja frecuencia (LFO):** Se utilizan como un elemento generador de señales para modular parámetros que se controlan por voltaje.

Generan la señal a la cual se le van a modificar parámetros como la amplitud o el contenido armónico para dar forma al sonido final.

Estos osciladores generan igual que los **VCO** señales básicas como senoidales, triangulares, cuadradas, y rampas, como las mostradas en la [Figura 2].

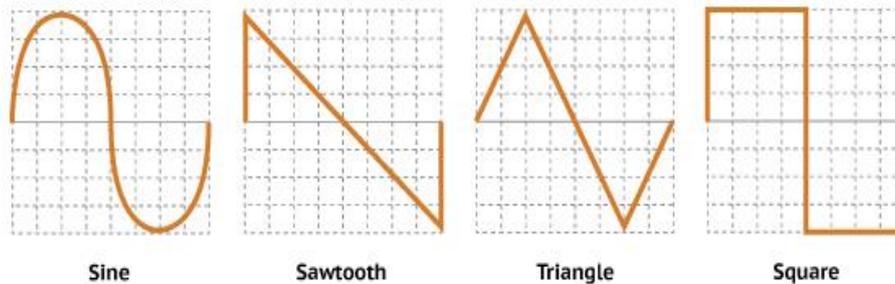


Figura 2 - Formas de onda más comunes

Estas señales son variables en frecuencia en el rango de muy baja frecuencia [0.02, 20]Hz.

Igual que los **VCO**, su frecuencia es ajustable por voltaje, permitiendo ajustar la frecuencia desde entradas analógicas.

- **Filtro controlado por voltaje (VCF):** Elemento que permite cambiar el contenido armónico de la señal que reciba en la entrada. Nos encontramos con varias clases de filtros dependiendo del contenido armónico que filtren:

Paso bajo: Permiten el paso de componentes de una frecuencia inferior a la frecuencia de corte, atenuando las componentes que queden por encima.

Paso alto: Permiten el paso de componentes de una frecuencia superior a la frecuencia de corte, atenuando las componentes que queden por debajo.

Paso banda: Permiten el paso de componentes de una frecuencia cercana a la frecuencia de corte, atenuando las componentes que se alejen de la misma.

En la [Figura 3] se encuentra una representación gráfica de su función.

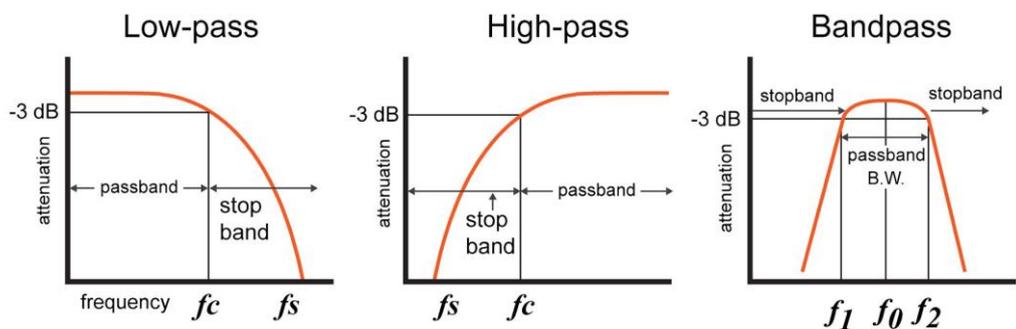


Figura 3 - Filtros más comunes

Estos filtros suelen incorporar un circuito que permite controlar la resonancia para amplificar las componentes de la señal que se encuentren en la frecuencia de corte del filtro.

La frecuencia de corte y resonancia son parámetros controlados por voltaje, permitiendo su ajuste desde entradas analógicas.

- **ADSR:** Este módulo se emplea para dar expresividad a los sonidos, permitiendo ajustar la forma de un control de voltaje respecto al tiempo.

Esto hace que las señales de control de voltaje adquieran una forma más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo. Este ajuste tiene 4 partes progresivas, mostradas también en la [Figura 4]:

Attack: tiempo que tardará el control de voltaje en alcanzar su nivel máximo empezando desde 0.

Decay: tiempo que tardará el control de voltaje en decaer al nivel sostenido desde el punto máximo.

Sustain: nivel del voltaje del control en el cual se queda sostenido hasta que comience la fase de release.

Release: tiempo que tardará el control de voltaje en decaer a 0 desde el voltaje sostenido.

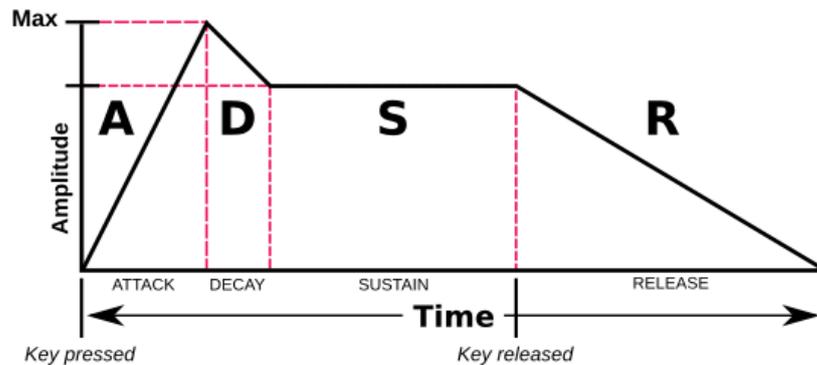


Figura 4 - Forma generada por la envolvente

Este módulo es para modular los controles de voltaje presentes en **VCAs**, **VCOs**, **LFOs** y **VCFs**.

- **Amplificador controlado por voltaje (VCA):** Elemento que permite modular la amplitud de la señal entrante. Su funcionamiento puede verse ilustrado en la [Figura 5]

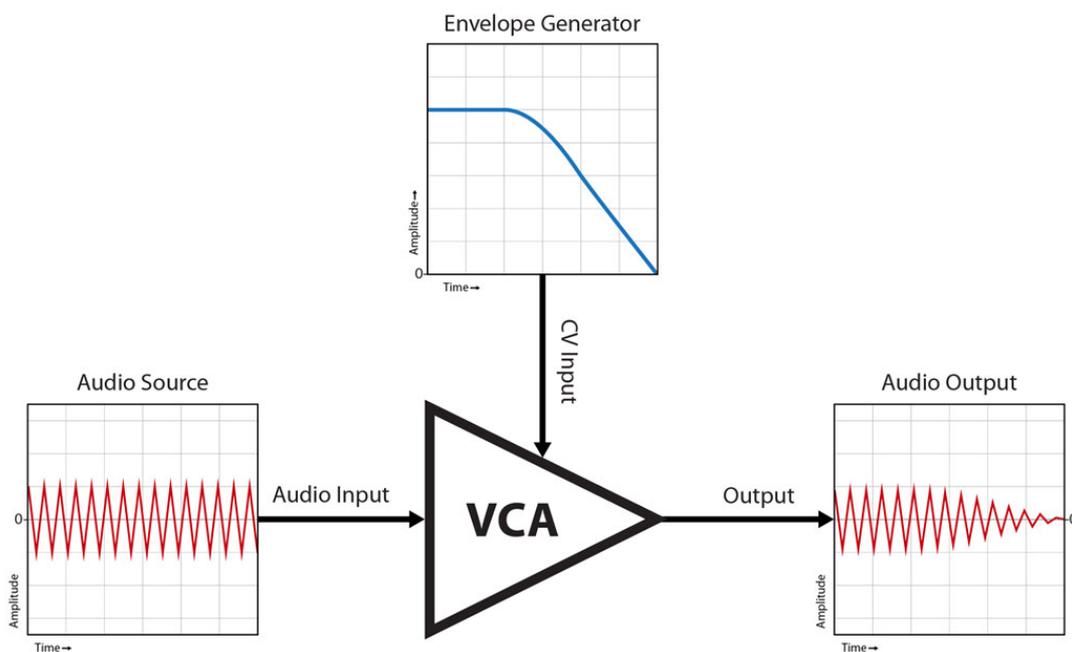


Figura 5 - Muestra del funcionamiento del VCA

Su ganancia depende del voltaje de la señal de control, permitiendo que la señal entrante sea amplificada o atenuada dependiendo de este voltaje.

La ganancia es un parámetro controlado por voltaje, permitiendo su ajuste desde entradas analógicas.

3. Análisis del mercado de sintetizadores modulares

3.1. Breve análisis de mercado:

El mercado de los sintetizadores es extenso debido a su gran cantidad de variedades, pudiendo ajustarse exactamente a lo que el cliente quiere, ya sea un productor musical o aficionado.

Esta gran variedad de dispositivos abarca un gran número de personas, pero nos vamos a centrar en el mercado que mueve la **síntesis sustractiva**, el propósito de nuestro dispositivo. El mercado de este tipo de síntesis está en buena forma, se puede observar por la gran cantidad de fabricantes que existen, y los miles de módulos que están a la venta. [9]

Actualmente en este mercado, se dispone de muchas opciones en cuanto a tipos de dispositivos en venta, abarcando ampliamente las necesidades de la gente interesada, por tanto, se procede a analizar las tendencias de este mercado:

- La síntesis modular ha tenido un gran crecimiento en los últimos años, atrayendo a los usuarios por su gran capacidad creativa.
- Los sintetizadores clásicos analógicos, tanto con reediciones de equipos de los 70s y 80s como los sintetizadores nuevos que integran posibilidades ampliadas, siguen estando presentes y gozan de buena salud.

Por tanto, se puede decir que el escenario no puede calificarse sino de positivo [20]. En cuanto a cifras, se puede hablar de las unidades vendidas de los sintetizadores más famosos [21]:

Korg M1	–	250.000	unidades
Roland D-50	–	175.000	unidades
Yamaha DX7	–	160.000	unidades
Roland SH-101	–	50.000	unidades
Ensoniq ESQ-1	–	50.000	unidades

3.2. Análisis del cliente:

La síntesis sustractiva es un tipo de síntesis que requiere conocer con exactitud los procesos que se le aplican a la señal para modularla, ya que hay que realizar complejos cableados entre módulos para obtener un sistema sustractivo. Por esto mismo, el público al que nos dirigimos es un público con **conocimientos de electrónica y sonido**, ya experimentados en la materia.

Especificando más, nuestro público son adultos generalmente, por su alto poder adquisitivo, esto es importante ya que la síntesis modular es cara debido a su complejidad, por tanto nuestros clientes tienen que disponer de dinero y ganas de gastarlo en sus aficiones.

Obviamente, nuestro público tiene que tener por afición la producción de sonido, más en concreto, afición por los sintetizadores modulares, que es en el mercado que estamos situados, ya que estos sistemas de síntesis sustractiva se basan en múltiples módulos.

Profundizando más en nuestro cliente medio, podríamos decir que es un hombre, debido a su gran superioridad a la mujer en número en esta afición. **Este cliente será de una edad comprendida entre 30 y 60 años, con un poder adquisitivo medio-alto, varón, y con la síntesis sustractiva como afición [10].**

Para promocionar la marca, se creó a mediados de marzo una página en Facebook. Esta página alcanza con sus publicaciones a miles de personas y tiene a mitad de Abril, 300 seguidores. Facebook nos proporciona información de los seguidores, y refuerzan nuestro análisis del cliente en la [Figura 6]:

Datos demográficos totales sobre las personas a las que les gusta tu página basados en la información de edad y sexo que proporcionan en sus perfiles de usuario.

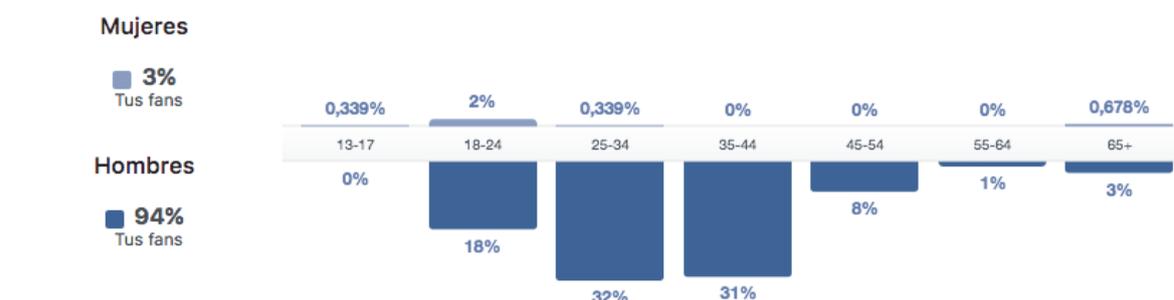


Figura 6 - Corroboración del análisis del cliente

3.3. Análisis de competencia:

Como he mencionado anteriormente, en este sector hay un gran número de fabricantes que diseñan y producen gran variedad de módulos. Dentro de estos fabricantes destacan **Mutable Instruments** [17], **Make Noise** [18] y **Xaoc Devices** [19], visiblemente son los preferidos de los consumidores por su popularidad.

Cabe diferenciar entre dos tipos de fabricantes:

- **Profesionales:** Empresas que se dedican enteramente al diseño y la fabricación de estos dispositivos. Los fabricantes mencionados anteriormente son de este tipo. Ofrecen dispositivos de alto nivel tanto en funcionalidades como en diseño. Nosotros nos disponemos a entrar en este sector, ya que queremos hacer de este proyecto un negocio viable.
- **Aficionados:** Particulares que diseñan y montan módulos que también son comercializados pero de formas menos profesionales, suelen emplear plataformas como Ebay o Tindie para comercializar sus módulos. Este sector es menos profesional ya que son diseños menos cuidados, y su imagen cara al público no está definida, por ello nos encasillamos en el otro tipo de fabricantes.

Entre fabricantes profesionales y aficionados, día a día se crean nuevos módulos que añaden funcionalidades o mejoran las ya existentes [11].

3.4. Tecnología disponible:

En el mundo de la síntesis modular no hay una preferencia común en cuanto a tecnología, es un sitio donde hay cabida para lo **analógico y lo digital**, ambos conviven hoy en día sin la predominancia de ninguno de los dos, digamos que existe un equilibrio entre ellos.

La síntesis modular emplea circuitos tanto analógicos como digitales, es el consumidor el que elige qué productos va a utilizar, ya que suele haber módulos de estas dos tecnologías con muy similares funcionalidades.

Por ello es un mundo de contrastes, ya que se siguen utilizando diseños analógicos de los años 70 como clones de los antiguos sintetizadores que marcaron esa época, mientras que se utiliza la última tecnología en cuanto a procesadores de señal de alta velocidad y diseños electrónicos basados en microcontroladores con alta capacidad de cálculo.

Este amplio campo de tecnología nos abre muchas posibilidades en cuanto al diseño de nuestros módulos. Nuestra elección es utilizar los dos campos, para ello vamos a clasificar nuestros módulos por su tecnología [1]:

- **Analógicos:** VCA y VCF son módulos basados en circuitos de naturaleza analógica. En este caso el circuito analógico obtiene ventajas respecto a su equivalente digital.
- **Digitales:** ADSR, VCO y LFO son módulos que utilizan tecnología digital para producir sus funciones. La elección de digital por encima de analógico reside en la capacidad de mejorar notablemente las funciones de un circuito analógico equivalente.

Por tanto en nuestro proyecto, vamos a trabajar estas dos tecnologías, permitiéndonos aprender sobre ambas.

3.5. Posibilidad de innovación en este mercado:

Este proyecto es una buena opción para innovar debido a la cantidad de tecnología existente en este ámbito, que nos permite no ser limitados en ningún momento.

Nuestra innovación en este mercado viene de la mano de implementar **nuevas funcionalidades** y dando más ajustes al cliente, haciendo que nuestro producto sea un dispositivo con unas funciones y utilidad superior a la media.

Esta es la diferenciación que buscamos, dar más funcionalidades por un precio justo. Estas mejoras están explicadas detalladamente en el apartado de **funciones del producto**.

3.6. Explicación de la elección de este mercado:

El mercado de la síntesis modular es un ecosistema adecuado para generar una repercusión con nuestros productos por diversos factores:

- **Perfil de comprador:** El público de este mercado, por norma general son personas de una edad media, comprendida entre [25-55] años, esto es un factor muy importante ya que son personas con una vida y trabajo estable que les permite poder gastar dinero en sus aficiones, en este caso la síntesis modular.
- **Aficionados a la música:** Las personas que forman parte de este mercado, son aficionados al sonido y la música, y hacen de ello su hobby. Al ser su hobby, no se suelen limitar a gastar dinero en algo que les facilita su proceso creativo a la hora de componer o crear sonido [12].
- **Conocen la electrónica:** Esta comunidad tiene conocimientos sobre electrónica, necesarios para entender exactamente qué procesos están ocurriendo en las señales para producir sonido. Esto es un factor positivo, ya que son gente que ya sabe lo que quiere, y no hace falta que tengas que explicarles con gran detalle el funcionamiento de lo que van a comprar, por tanto saben perfectamente que es lo que están comprando.
- **Comunidad DIY:** Al tener conocimientos sobre electrónica, hay gente que se anima a crear ellos mismos sus propios dispositivos (Como yo). Esto es muy positivo e interesante, ya que existe un continuo desarrollo de productos, seguido de un valorado feedback de sus usuarios, que te ayudan a perfeccionar tu dispositivo [13].

Estos son los factores más influyentes sobre este mercado, los que me han hecho decantarme por él.

3.7. Conclusiones:

En este capítulo hemos podido comprender qué es lo que me ha motivado a formar parte del mercado de la síntesis modular. Conocer a nuestro cliente es un proceso muy importante a la hora de comercializar algo, ya que tenemos que centrar nuestros esfuerzos en este sector determinado.

Una vez conocido el cliente, tenemos que conocer a la competencia, también es fundamental ya que vamos a competir por lo mismo, vender en este mercado. Esto nos servirá para mejorar las funcionalidades ya existentes, y dar precios competitivos.

La tecnología presente en el mercado es importante, ya que nos indica nuestras posibilidades de innovación. Justamente como he comentado anteriormente, este mercado nos da mucha libertad tecnológica.

En conclusión, el mercado de la síntesis modular me parece un sector perfecto para desarrollar dispositivos.

4. Especificación de requisitos

4.1. Introducción:

En esta sección se va a explicar las necesidades y propósitos que van a tener los módulos. Como se ha visto anteriormente, cada módulo tiene su función y sus requerimientos, en este apartado se analizará cada caso específico.

4.2. Propósito:

El objetivo planteado para este proyecto es la **creación de un sistema básico de síntesis sustractiva**, esto se conseguirá por medio de los diferentes módulos planteados en este documento.

El principal reto del proyecto será el mejorar las funcionalidades existentes en los módulos comercializados.

En este proyecto, nos planteamos mejoras respecto a los módulos equivalentes a nuestros diseños (VCO, LFO, VCA, ADSR y VCF) Estas mejoras nos ayudarán a despuntar por encima de los demás módulos, generando interés sobre los nuestros.

4.3. Ámbito del Sistema:

El ámbito de este sistema es crear un dispositivo completo de síntesis sustractiva. Este sistema se compone de módulos como **VCO, LFO, VCA, VCF y ADSR**.

Por tanto el sistema es capaz de crear señales fundamentales y modificar su contenido armónico y amplitud. Estos módulos se conectarán a una caja que los agarre a raíles mediante tornillos, esta caja proveerá también la alimentación necesaria para estos.

Para poder interactuar con los módulos se han diseñado con potenciómetros accesibles al usuario, y para interconexionar estos módulos se utilizan cables jack de 3.25mm mono.

En conclusión, este sistema se basa en módulos con funcionalidades diversas, una caja que los aloje, una fuente de alimentación para aportar la energía necesaria para su funcionamiento, y cables jack para conectar entre sí las señales de audio y de control.

4.4. Definición de los requisitos:

Los requisitos del sistema están divididos en dos categorías principales, ya que son los requisitos más importantes:

- **Requisitos de rendimiento:** El rendimiento es un factor muy determinante a la hora de elegir un microcontrolador para diseñar un dispositivo. En este caso, los módulos digitales (VCO, LFO, Envolvente) exigen diferentes rendimientos, por tanto hay que ajustarse a lo que cada uno necesite.
- **Requisitos de diseño:** Los módulos diseñados se ajustan a un protocolo de sintetizadores modulares llamado **Eurorack** [14]. Este protocolo dicta las especificaciones mecánicas y de alimentación.

Estos requisitos son ampliados y detallados en la siguiente sección de **Requisitos específicos**.

4.5. Definiciones, acrónimos y abreviaturas:

Este documento incluye un gran número de acrónimos que pueden no ser comprendidos, así que en este apartado los definimos:

- **VCO:** (Voltage Controlled Oscillator) Módulo creado para este proyecto consistente en un oscilador que funciona a frecuencia audible. Es el corazón de este sistema de síntesis. Sus parámetros se controlan con voltaje.
- **LFO:** (Low Frequency Oscillator) Módulo creado para este proyecto consistente en un oscilador que funciona a baja frecuencia. Se utiliza para modular y variar parámetros controlables por voltaje. Sus parámetros se controlan con voltaje.
- **VCF:** (Voltage Controlled Filter) Módulo creado para este proyecto consistente en un filtro controlado por voltaje que permite cambiar el contenido armónico de la señal de entrada. Sus parámetros se controlan con voltaje.
- **VCA:** (Voltage Controlled Amplifier) Módulo creado para este proyecto consistente en un amplificador controlado por voltaje que permite cambiar la amplitud de la señal de entrada. Sus parámetros se controlan con voltaje.
- **ADSR:** Este módulo se emplea para dar expresividad a los sonidos, esto hace que las señales de control de voltaje adquieran una forma

más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo.

5. Descripción general

El dispositivo tiene que ofrecer al usuario un sistema completo de síntesis de audio, esto quiere decir que los módulos incluidos tienen que formar un equipo básico comprendido por osciladores, filtros y amplificadores, además de ofrecer controles por voltaje para variar los parámetros de cada uno, ya sea frecuencia, resonancia o cualquier otro parámetro ajustable.

5.1. Funciones del producto:

Sintetizando la breve descripción anterior, vamos a organizar el producto en torno a varias funcionalidades:

- **VCO:** Generar una señal oscilante de una frecuencia comprendida entre [20-1000]Hz y de amplitud [-5,5]V, ajustable en frecuencia y forma de onda.
- **LFO:** Generar una señal oscilante de una frecuencia comprendida entre [0.01-100]Hz de amplitud en la salida unipolar [0,5]V y en la bipolar [-2.5,2.5]V ajustable en frecuencia y forma de onda.
- **VCF:** Filtrar la señal de audio en torno a frecuencias comprendidas entre [20-20000]Hz, pudiendo ajustar la resonancia y frecuencia de corte del filtro y tener dos salidas simultáneas, Low Pass y Band Pass.
- **VCA:** Amplificar o atenuar la señal con control de voltaje, ajustando la ganancia requerida.
- **ADSR:** Generar una señal envolvente para dar expresividad a los sonidos, hacer que las señales de control de voltaje adquieran una forma más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo, con posibilidad de retrigger.

5.2. Características de los usuarios:

Las personas que utilicen nuestro producto son personas con conocimientos en sonido, electrónica, y producción musical. Esto es un perfil bastante específico, por tanto nos permite centrar nuestros esfuerzos para hacer algo realmente innovador y provechoso para este grupo de aficionados.

Concretamente, las personas que usen nuestros productos tienen que tener conocimientos en síntesis modular o estar aprendiendo sobre ella, ya que los módulos son para este sistema.

Por tanto, nuestros usuarios son personas que están buscando módulos para sus sintetizadores, son conocedores del funcionamiento de estos y buscan en nuestros módulos funcionalidades superiores a la media.

5.3. Restricciones:

Nuestro sistema de síntesis modular depende de un **soporte** con las especificaciones mecánicas del formato Eurorack, esto quiere decir que necesita de dos raíles de sujeción para montarlos. Estos raíles tienen que contar con tuercas de métrica M3 para sujetar los módulos con tornillos a ellos. La distancia entre los raíles es equivalente a 3U (133.4mm).

A parte de estos raíles, también necesitan estos módulos una **fuentes de alimentación**. Cualquier fuente especificada para Eurorack cumpliría esta función. En este caso utilizamos la fuente de alimentación de **TipTop Audio, uZeus** [16], ya que genera el voltaje y corriente adecuado para nuestro sistema.

Estas restricciones son comentadas más exhaustivamente en la siguiente sección de **Requisitos específicos**.

6. Requisitos específicos.

Sección donde se profundiza en los requisitos del sistema de síntesis modular, ya sean requisitos de rendimiento, de diseño, atributos mecánicos y otros requisitos.

6.1. Requisitos de rendimiento:

El rendimiento es un factor muy determinante a la hora de elegir un microcontrolador para diseñar un dispositivo. En este caso, los módulos digitales (VCO, LFO, ADSR) exigen diferentes rendimientos, por tanto hay que ajustarse a lo que cada uno necesite.

- **VCO:** Este módulo tiene una alta carga computacional, ya que generar ondas sonoras de diferentes formas y a una alta frecuencia, es un reto. Para cumplir estos objetivos, se va a utilizar un microcontrolador **STM32F3**, ya que incluye un núcleo Cortex M4, con un rendimiento adecuado y suficiente frecuencia del reloj para esta aplicación.
- **LFO:** Módulo muy similar al VCO, tiene una alta carga computacional porque genera ondas de baja frecuencia pero con una resolución muy alta. Para cumplir estos objetivos, se va a utilizar un microcontrolador **STM32F3** como el empleado en el módulo VCO, ya que incluye un núcleo Cortex M4, con un rendimiento adecuado y suficiente frecuencia del reloj para esta aplicación.
- **ADSR:** Módulo que se emplea para dar expresividad a los sonidos, esto hace que las señales de control de voltaje adquieran una forma más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo. Esta aplicación no requiere una alta velocidad computacional, por tanto vamos a diseñar este módulo basándonos en el microcontrolador **ATMEGA328**, conocido por ser el utilizado por el Arduino Uno. Este microcontrolador no es tan potente como el STM32 utilizado en el **VCO** y **LFO**, pero se ajusta a nuestras necesidades ya que funciona a 16MHz, suficiente para esta función.

Por tanto, los requisitos de rendimiento se basan en la potencia requerida del microcontrolador utilizado.

6.2. Restricciones de diseño:

Los módulos diseñados se ajustan a un protocolo de sintetizadores modulares llamado **Eurorack** [14]. Este protocolo dicta las especificaciones mecánicas y de alimentación, desarrolladas en los siguientes puntos:

- **Restricciones mecánicas:** Los módulos tendrán 128.5mm de altura para ajustarse a los raíles de sujeción. Estos módulos tendrán agujeros para colocar los tornillos necesarios para apretar los módulos a los raíles, los tornillos son de métrica M3 y deben de estar situados a 3mm del final del panel en la vertical, y a 7.5mm del final del panel en la horizontal, como lo indica esta [Figura 7]:

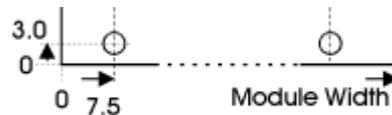


Figura 7 - Posicionamiento de los agujeros de sujeción al raíl

Debido a los raíles, la PCB no puede exceder los 110mm de longitud vertical ya que en caso de superarlo, no podría montarse el módulo en los raíles.

- **Restricciones de alimentación:** Los módulos se alimentan desde una fuente de alimentación que cumpla con el formato Eurorack [15]. Esto quiere decir que la fuente ofrece estos voltajes DC: **+12V, +5V, GND, -12V**.

A parte de los voltajes requeridos, también es una restricción el conector que utiliza tanto la fuente como el módulo. Para adaptarnos a la solución más común, se ha elegido utilizar conectores de 16 pines para la fuente de alimentación, y de 10 pines para el módulo, siguiendo el patillaje de la [Figura 8]:

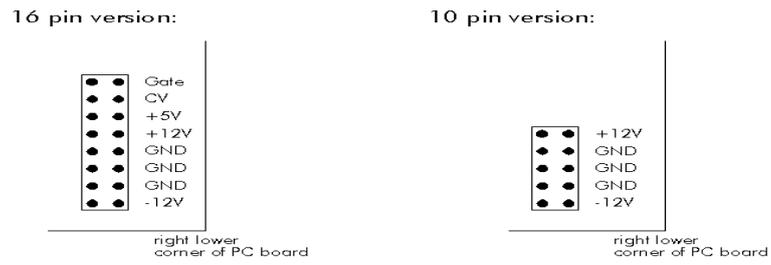


Figura 8 - Asignación de pines de los conectores Eurorack

Al tener un patillaje en la fuente de alimentación y otro diferente en el módulo, el cable que una ambos tendrá que ajustarse a este requisito, por tanto el cable utilizado para conectar los módulos será de 16 pines en un conector y 10 en el otro, exactamente como el de esta [Figura 9]:

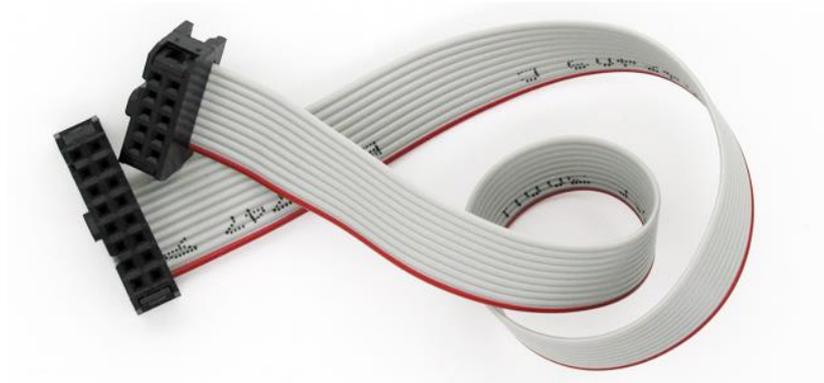


Figura 9 - Conector de alimentación Eurorack

Por tanto, las restricciones de diseño de los módulos vienen determinadas tanto por restricciones mecánicas como de alimentación.

6.3. Atributos mecánicos del sistema:

El sistema se basa como se ha comentado anteriormente, en el estándar Eurorack [14], esto define en parte los atributos mecánicos, ya que se sujeta a unos raíles con unas dimensiones específicas.

- Los módulos tendrán **128.5mm de altura** para ajustarse a los raíles de sujeción.
- Cada módulo tendrá **30mm de ancho**, espacio justo para poder incluir los componentes de cada uno.
- También compartirán los **40mm de profundidad** aproximadamente contando los elementos mecánicos que permiten su ajuste.

Por tanto, cada módulo tendrá estas dimensiones: **128.5x30x40mm**. Considerando que los módulos tienen que ser montados uno al lado del otro, podemos calcular la longitud de raíl necesaria para montar un sistema básico compuesto por un módulo de cada tipo (VCO, VCF, VCA, LFO y ADSR).

$$\text{Longitud Raíl} = \text{número de módulos} * \text{ancho de cada módulo} = 5 * 30\text{mm} = 150\text{mm}$$

Estos 150mm sería la longitud mínima de raíl para integrar los 5 módulos elementales, no obstante, se dispone de unos raíles mucho más largos, por tanto no existen problemas de espacio.

Conociendo las dimensiones de los módulos, se ha procedido a fabricar una caja que los albergue y alimente, para así contar con espacio de sobra por si se fabrican más módulos. Las dimensiones de esta caja fabricada en madera son: **450x280x100mm**.

Esta caja contiene dos pares de raíles Z-Rails de Tip Top Audio [22], de 426mm, permitiendo el montaje de dos filas de módulos independientes. La apariencia de esta caja está en la [Figura 10]



Figura 10 - Vista en Solidworks de la caja formato Eurorack.

En conclusión, el sistema ocupará **450x280x100mm**, y habrá espacio disponible para el montaje de más módulos si se requiere.

7. Entorno de desarrollo.

7.1. Introducción:

Este sistema contiene un desarrollo tecnológico extenso, el orden se va a articular conforme al natural desarrollo del proyecto, empezando por el diseño del circuito electrónico (**Hardware**), y después el desarrollo digital (**Software**).

7.2. Tecnologías utilizadas:

Vamos a diferenciar en este apartado entre el **hardware** y el **software**, ya que usan entornos de desarrollo completamente diferentes, aplicados a su función.

7.2.1. Hardware:

Este apartado se compone del diseño del esquema electrónico y el posterior diseño de PCB. Para realizar estas funciones, se emplea el software de diseño **KiCad**, ya que nos permite realizar exactamente lo que necesitamos: Primero un diseño del esquema eléctrico, y luego la PCB.

KiCad: Es un software gratuito para diseños electrónicos. Facilita la realización de esquemas para los circuitos electrónicos y su conversión a PCBs. KiCad fue desarrollado por Jean-Pierre Charras. Incluye un entorno de desarrollo para los esquemáticos y para diseño de circuitos impresos. Entre sus herramientas están presentes, dispone de generador de archivos Gerber y vistas 3D [23].

7.2.2. Software:

En cuanto a software, hay que diferenciar entre los dos microcontroladores digitales que se emplean en los módulos. Por una parte, el módulo basado en ATmega328 (Envelope) se desarrolla en el entorno **Arduino**, que incorpora las funcionalidades básicas para compilar y programar este tipo de microcontroladores. Por la otra parte, tenemos módulos basados en STM32 (VCO, LFO), estos se desarrollan en el entorno **Keil**, que nos proporciona el compilador, programador y depurador para la familia de dispositivos STM32.

Arduino: Es un entorno de desarrollo integrado, implementado en el lenguaje de programación Java. Incluye un editor de código con diferentes funciones, como corte y pegado, búsqueda y reemplazo... Se diferencia por incluir un mecanismo de tan solo un click para compilar y subir programas a cualquier placa que disponga de un ATmega328. También cuenta con un área para la consola, y herramientas [24].

Keil: Es un compilador de lenguaje C, que incluye una gran cantidad de herramientas de desarrollo, como compilador de ANSI C, depurador, simulador y gestor de librerías. Además da soporte a la familia de microcontroladores STM32, siendo ideal para los módulos de el sistema [25].

8. Diseño.

8.1. Introducción:

En este apartado, se procede a explicar con detalle el proceso de diseño y desarrollo seguido por los módulos del sistema, empezando por el **Software** si dispone de el mismo, después se mostrará el **Hardware**, y para finalizar se verá el diseño de los componentes mecánicos realizados para albergar estos módulos en el sistema Eurorack [14].

8.2. Entradas y salidas:

Los módulos necesitan de unas entradas y salidas para su operación en el sistema, es por ello que se ha considerado muy importante explicar sus entradas, salidas y la naturaleza de las mismas. Los módulos presentes en el sistema serán analizados en este apartado:

- **VCO** : Es la fuente de sonido en este sistema de síntesis sustractiva. Este oscilador genera la señal a la cual se le van a modificar parámetros como la amplitud o el contenido armónico para dar forma al sonido final.

Este oscilador, tiene los siguientes ajustes para permitir una alta adaptación a las exigencias:

- **Conectores Jack:** Conectores para cables Jack monoaurales de 3.5mm de diámetro, estos permiten ajustar parámetros del módulo desde un módulo externo.
 - **Jack de entrada 1 Voltio / Octava:** Conector empleado para la entrada de voltaje con el protocolo 1 Voltio / Octava [26]. Esta entrada es la que se conecta a teclados o controladores que se encargan de proveer al oscilador de un voltaje que altere la frecuencia de la señal de salida para producir sonidos armónicos.
 - **Jack de entrada de modulación de frecuencia:** Conector empleado para la entrada de voltaje, encargada de proveer al oscilador de un voltaje que altere la frecuencia de la señal de salida para producir sonidos armónicos. La diferencia con la entrada 1V/O es que esta sigue un comportamiento lineal, además de tener un potenciómetro atenuador para ajustar su comportamiento.
 - **Jack de entrada de forma de onda:** Conector empleado para cambiar la forma de onda de la señal de salida entre las disponibles (Senoidal, cuadrada, triangular...) Este conector

recibe señales de tipo trigger, o pulsos. Se complementa con el botón utilizado para esta misma función.

- **Jack de salida de audio:** Conector empleado para la salida de la señal de audio, consistente en una señal de 10 Voltios Pico Pico de amplitud, y de una frecuencia entre [20-1000]Hz.
- **Potenciómetros:** Mecanismos de ajuste que con el giro del mismo permiten la configuración de algunos parámetros de este módulo.
 - **2 Potenciómetros de ajuste de frecuencia:** Estos potenciómetros embebidos en el sistema sirven para ajustar de forma precisa la frecuencia de oscilación. 1 Potenciómetro se encarga del ajuste fino, y el otro del grueso, cubriendo así todo el espectro de frecuencias que vamos a emplear.
 - **1 Potenciómetro de ajuste de FM (Frecuencia Modulada):** Este potenciómetro es simplemente un atenuador para la señal que proviene de la entrada de modulación de frecuencia.
- **Botones:** Mecanismos de ajuste que permiten la configuración de algunos parámetros de este módulo debido a la lectura de su estado.
 - **1 Botón de ajuste de forma de onda:** Pulsador empleado para cambiar la forma de onda de la señal de salida entre las disponibles (Senoidal, cuadrada, triangular...) Este botón se complementa con el conector Jack utilizado para esta misma función.
- **Indicadores visuales:** Indicadores luminosos para identificar diferentes funciones en el módulo.
 - **1 LED RGB de visualización de la forma de onda:** Indicador empleado para identificar la forma de onda de la señal de salida, ya que está asociada a un color presente en este led.
- **LFO :** Se utilizan como un elemento generador de señales para modular parámetros que se controlan por voltaje. Igual que los **VCO**, su frecuencia es ajustable por voltaje, permitiendo ajustar la frecuencia desde entradas analógicas.
- **Conectores Jack:** Conectores para cables Jack monoaurales de 3.5mm de diámetro, estos permiten ajustar parámetros del módulo desde un módulo externo.

- **Jack de entrada de modulación de frecuencia:** Conector empleado para la entrada de voltaje, encargada de proveer al oscilador de un voltaje que altere la frecuencia de la señal de salida. Esta entrada dispone de un potenciómetro atenuador para ajustar su comportamiento.
 - **Jack de entrada de forma de onda:** Conector empleado para cambiar la forma de onda de la señal de salida entre las disponibles (Senoidal, cuadrada, triangular...) Este conector recibe señales de tipo trigger, o pulsos. Se complementa con el botón utilizado para esta misma función.
 - **Jack de salida unipolar:** Conector empleado para la salida de la señal de control de voltaje, consistente en una señal de voltaje comprendido entre **[0 / 5] V**, y de una frecuencia entre [0.01-100]Hz.
 - **Jack de salida bipolar:** Conector empleado para la salida de la señal de control de voltaje, consistente en una señal de voltaje comprendido entre **[-2.5 / 2.5] V**, y de una frecuencia entre [0.01-100]Hz.
- **Potenciómetros:** Mecanismos de ajuste que con el giro del mismo permiten la configuración de algunos parámetros de este módulo.
- **1 Potenciómetro de ajuste de frecuencia:** Este potenciómetro embebido en el sistema sirve para ajustar de forma precisa la frecuencia de oscilación.
 - **1 Potenciómetro de ajuste de FM (Frecuencia Modulada):** Este potenciómetro es simplemente un atenuador para la señal que proviene de la entrada de modulación de frecuencia.
- **Botones:** Mecanismos de ajuste que permiten la configuración de algunos parámetros de este módulo debido a la lectura de su estado.
- **1 Botón de ajuste de forma de onda:** Pulsador empleado para cambiar la forma de onda del voltaje de control de salida entre las disponibles (Senoidal, cuadrada, triangular...) Este botón se complementa con el conector Jack utilizado para esta misma función.
 - **1 Botón de ajuste de frecuencia de onda:** Pulsador empleado para cambiar el rango de la frecuencia de la señal

del voltaje de control de salida entre las disponibles (Muy lento, Operación normal, Rápido).

- **Indicadores visuales:** Indicadores luminosos para identificar diferentes funciones en el módulo.
 - **1 LED RGB de visualización de la forma de onda:** Indicador empleado para identificar la forma de onda de la señal de salida y su amplitud.
- **VCF :** Elemento que permite cambiar el contenido armónico de la señal que reciba en la entrada. Su frecuencia de corte es ajustable por voltaje, también la resonancia, permitiendo los ajustes desde entradas analógicas.
 - **Conectores Jack:** Conectores para cables Jack monoaurales de 3.5mm de diámetro, estos permiten ajustar parámetros del módulo desde un módulo externo.
 - **Jack de entrada de modulación de frecuencia:** Conector empleado para la entrada de una señal encargada de proveer al filtro de un voltaje que altere la frecuencia de corte del filtro. Esta entrada dispone de un potenciómetro atenuador e inversor para ajustar su comportamiento.
 - **Jack de entrada de modulación de frecuencia con protocolo 1 Voltio / Octava:** Conector empleado para la entrada de una señal encargada de proveer al filtro de un voltaje que altere la frecuencia de corte del filtro. Esta entrada sigue el protocolo 1V/O que permite ajustar su frecuencia de corte de forma exponencial.
 - **Jack de entrada de modulación de resonancia:** Conector empleado para la entrada de una señal encargada de proveer al filtro de un voltaje que altere la resonancia del filtro.
 - **Jack de entrada de señal:** Conector empleado para la entrada de la señal de audio.
 - **Jack de salida paso bajo de señal:** Conector empleado para la salida de la señal de audio ya filtrada mediante el filtro paso bajo.
 - **Jack de salida paso banda de señal:** Conector empleado para la salida de la señal de audio ya filtrada mediante el filtro paso banda.

- **Potenciómetros:** Mecanismos de ajuste que con el giro del mismo permiten la configuración de algunos parámetros de este módulo.
 - **1 Potenciómetro de ajuste de frecuencia:** Este potenciómetro embebido en el sistema sirve para ajustar de forma precisa la frecuencia de corte del filtro.
 - **1 Potenciómetro de ajuste de FM (Frecuencia Modulada):** Este potenciómetro es simplemente un atenuador e inversor para la señal que proviene de la entrada de modulación de frecuencia.
 - **1 Potenciómetro de ajuste de resonancia:** Este potenciómetro embebido en el sistema sirve para ajustar de forma precisa la resonancia del filtro.

- **VCA :** Elemento que permite modular la amplitud de la señal entrante. Su ganancia depende del voltaje de la señal de control, permitiendo que la señal entrante sea amplificada o atenuada dependiendo de este voltaje. La ganancia es un parámetro controlado por voltaje, permitiendo su ajuste desde entradas analógicas.

Se debe tener en cuenta que este módulo dispone de 2 VCAs, por tanto se va a exponer tan solo los componentes de 1 VCA.

- **Conectores Jack:** Conectores para cables Jack monoaurales de 3.5mm de diámetro, estos permiten ajustar parámetros del módulo desde un módulo externo.
 - **Jack de entrada de señal de audio de frecuencia:** Conector empleado para la entrada de la señal de audio.
 - **Jack de entrada de señal de audio de frecuencia:** Conector empleado para la salida de la señal de audio ya modificada en amplitud.
 - **Jack de entrada de señal de modulación de amplitud 1:** Conector empleado para la entrada de una señal de control de voltaje para modificar la ganancia de la señal de salida. Esta entrada pasa por un atenuador.
 - **Jack de entrada de señal de modulación de amplitud 2:** Conector empleado para la entrada de una señal de control de voltaje para modificar la ganancia de la señal de salida complementario a la otra entrada de modulación.

- **Potenciómetros:** Mecanismos de ajuste que con el giro del mismo permiten la configuración de algunos parámetros de este módulo.
 - **1 Potenciómetro de ajuste de ganancia:** Este potenciómetro embebido en el sistema sirve para ajustar de forma precisa la ganancia del VCA.
 - **1 Potenciómetro de ajuste de modulación:** Este potenciómetro es simplemente un atenuador para la señal que proviene de la entrada de modulación de amplitud 1.

- **Indicadores visuales:** Indicadores luminosos para identificar diferentes funciones en el módulo.
 - **1 LED de visualización de la ganancia** Indicador empleado para identificar el nivel de la señal de salida.

- **Envolvente :** Este módulo se emplea para dar expresividad a los sonidos, permitiendo ajustar la forma de un control de voltaje respecto al tiempo. Esto hace que las señales de control de voltaje adquieran una forma más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo.

- **Conectores Jack:** Conectores para cables Jack monoaurales de 3.5mm de diámetro, estos permiten ajustar parámetros del módulo desde un módulo externo.
 - **Jack de entrada de gate:** Conector empleado para la entrada de una señal gate, encargada de iniciar el procesado de la señal envolvente.
 - **Jack de entrada de retrigger:** Conector empleado para la entrada de una señal trigger, o pulso, encargada de reiniciar el procesado de la señal envolvente mientras está ya en proceso.
 - **Jack de salida:** Conector empleado para la salida de la señal de control de voltaje, consistente en una señal de voltaje comprendido entre **[0 / 5] V**.

- **Potenciómetros:** Mecanismos de ajuste que con el giro del mismo permiten la configuración de algunos parámetros de este módulo.

- **1 Potenciómetro de ajuste de attack:** Este potenciómetro embebido en el sistema sirve para ajustar de forma precisa el tiempo de subida de la señal de salida.
 - **1 Potenciómetro de ajuste de decay:** Este potenciómetro embebido en el sistema sirve para ajustar de forma precisa el tiempo de bajada hasta el nivel sostenido de la señal de salida.
 - **1 Potenciómetro de ajuste de sustain:** Este potenciómetro embebido en el sistema sirve para ajustar de forma precisa el nivel sostenido de la señal de salida mientras exista un gate en la entrada.
 - **1 Potenciómetro de ajuste de release:** Este potenciómetro embebido en el sistema sirve para ajustar de forma precisa el tiempo de bajada de la señal de salida después de desaparecer la señal gate de la entrada.
- **Botones:** Mecanismos de ajuste que permiten la configuración de algunos parámetros de este módulo debido a la lectura de su estado.
 - **1 Botón de ajuste de tiempo:** Pulsador empleado para cambiar el tiempo de respuesta de la señal de salida ante la entrada gate, pudiendo ser ajustado entre una respuesta rápida o lenta.
 - **Indicadores visuales:** Indicadores luminosos para identificar diferentes funciones en el módulo.
 - **1 LED de visualización de la señal de salida:** Indicador empleado para identificar la forma de onda de la señal de salida y su amplitud.

8.3. Software:

De los 5 módulos que se compone este proyecto, solo 3 módulos son de naturaleza digital, siendo el **VCO**, **LFO** y **ADSR**. Esto quiere decir que estos 3 módulos son los que van a ser analizados detalladamente en este apartado.

Para estos módulos se va a seguir el mismo procedimiento de presentación, comenzando con una breve descripción de sus requerimientos, después un diagrama de bloques que aclare el funcionamiento del código, seguido de una explicación del mismo para que de un vistazo pueda entenderse su utilización.

VCO: Su función es generar una señal oscilante de una frecuencia comprendida entre [20-1000]Hz y de amplitud [-5,5]V, ajustable en frecuencia y forma de onda.

Para lograr este cometido, vamos a integrar el **STM32F334K8** en el sistema, ya que este microcontrolador tiene potencia suficiente para alcanzar los objetivos de este módulo. Para este dispositivo se necesita un gran número de **recursos del microcontrolador** para conseguir una operación correcta:

- **Timers:** Son muy importantes en este caso, ya que el Timer 2 se encarga de actualizar la salida del DAC a la frecuencia requerida, mientras que el Timer 3 actualiza el color y brillo del led RGB.
- **Interrupciones:** Necesarias para hacer una buena gestión de las lecturas del botón y entradas externas, evitan perder el tiempo leyendo en cada momento los estados de las entradas. Se utiliza para cambiar la forma de onda cuando se produzca un cambio en el estado del pin de entrada de forma de onda.
- **Entradas digitales:** Este módulo tan solo cuenta con una entrada digital, asociada a la selección de forma de onda, esta lleva una interrupción para aumentar la eficiencia del código.
- **Entradas analógicas:** Necesarias para hacer una lectura de los valores de los potenciómetros para ajustar la frecuencia, así como la entrada de 1V/O y FM.
- **Salida analógica:** Su función es generar la onda oscilante de frecuencia controlada.

Diagrama de bloques del código:

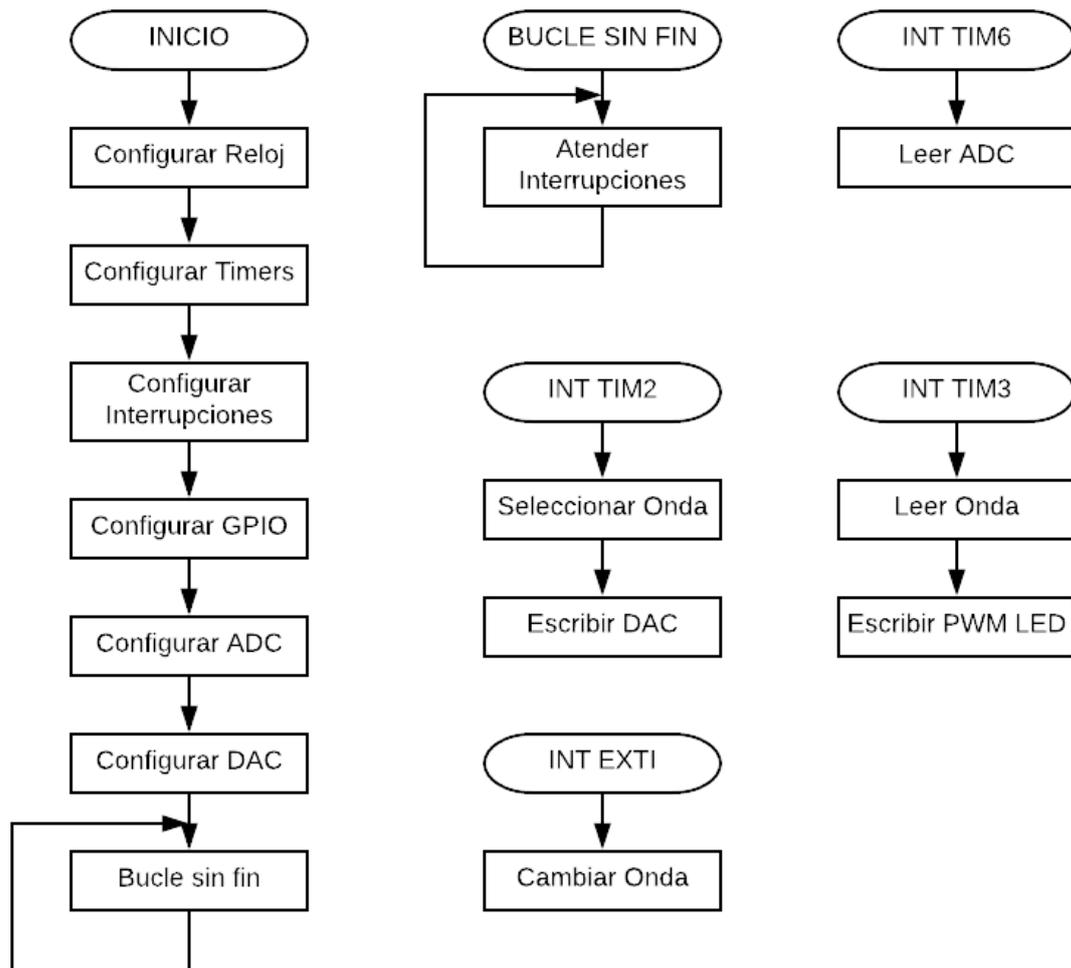


Figura 11 - Diagrama de bloques del código del VCO.

Explicación del diagrama de bloques:

En la parte izquierda de la [Figura 11], se puede observar todo el proceso que sigue el microcontrolador desde que se inicia hasta que funciona en bucle sin fin. En cuanto al inicio, lo que hace el microcontrolador es **configurar y activar** los periféricos para asegurar que funcione de forma correcta el sistema, este paso solo se realiza una vez.

Cuando la configuración llega a su fin, se llega al **bucle sin fin**, esto quiere decir que el microcontrolador no saldrá de este estado a no ser que se reinicie o apague. En este estado sin fin es donde se desarrolla toda la actividad requerida del VCO:

- El bucle sin fin no recorre ninguna función, tan solo espera recibir las interrupciones para ejecutar código.

- Timer6 tiene una interrupción a 100Hz, y cuando salta procede a la lectura de las entradas analógicas y registrar su valor.
- Timer3 es la interrupción a 100Hz que gestiona la iluminación y color del led RGB de la placa.
- Timer2 es la interrupción que se encarga de manejar la frecuencia de la escritura en el DAC.
- Se atiende a la interrupción externa de cambio de forma de onda.
- La interrupción externa del botón de forma de onda está conectada a una entrada de jack de tipo trigger, así que cualquier evento en este botón o jack cambia la forma de onda de la señal de salida.

LFO: Genera una señal oscilante de una frecuencia comprendida entre [0.01-100]Hz de amplitud en la salida unipolar [0,5]V y en la bipolar [-2.5,2.5]V ajustable en frecuencia y forma de onda.

Para lograr este cometido, vamos a integrar el **STM32F334K8** en el sistema, ya que este microcontrolador tiene potencia suficiente para alcanzar los objetivos de este módulo. Para este dispositivo se necesita un gran número de **recursos del microcontrolador** para conseguir una operación correcta:

- **Timers:** Son muy importantes en este caso, ya que el Timer 2 se encarga de actualizar la salida del DAC a la frecuencia requerida, mientras que el Timer 3 actualiza el color y luminosidad del led RGB.
- **Interrupciones:** Necesarias para hacer una buena gestión de las lecturas del botón y entradas externas, evitan perder el tiempo leyendo en cada momento los estados de las entradas. Se utilizan para cambiar la forma de onda y para ajustar el rango de frecuencias.
- **Entradas digitales:** Este módulo cuenta con dos entradas digitales, una asociada a la selección de forma de onda, y la otra a la selección del rango de frecuencias de ajuste. Ambas llevan una interrupción para aumentar la eficiencia del código.
- **Entradas analógicas:** Necesarias para hacer una lectura de los valores de los potenciómetros para ajustar la frecuencia y FM.
- **Salida analógica:** Su función es generar la onda oscilante de frecuencia controlada.

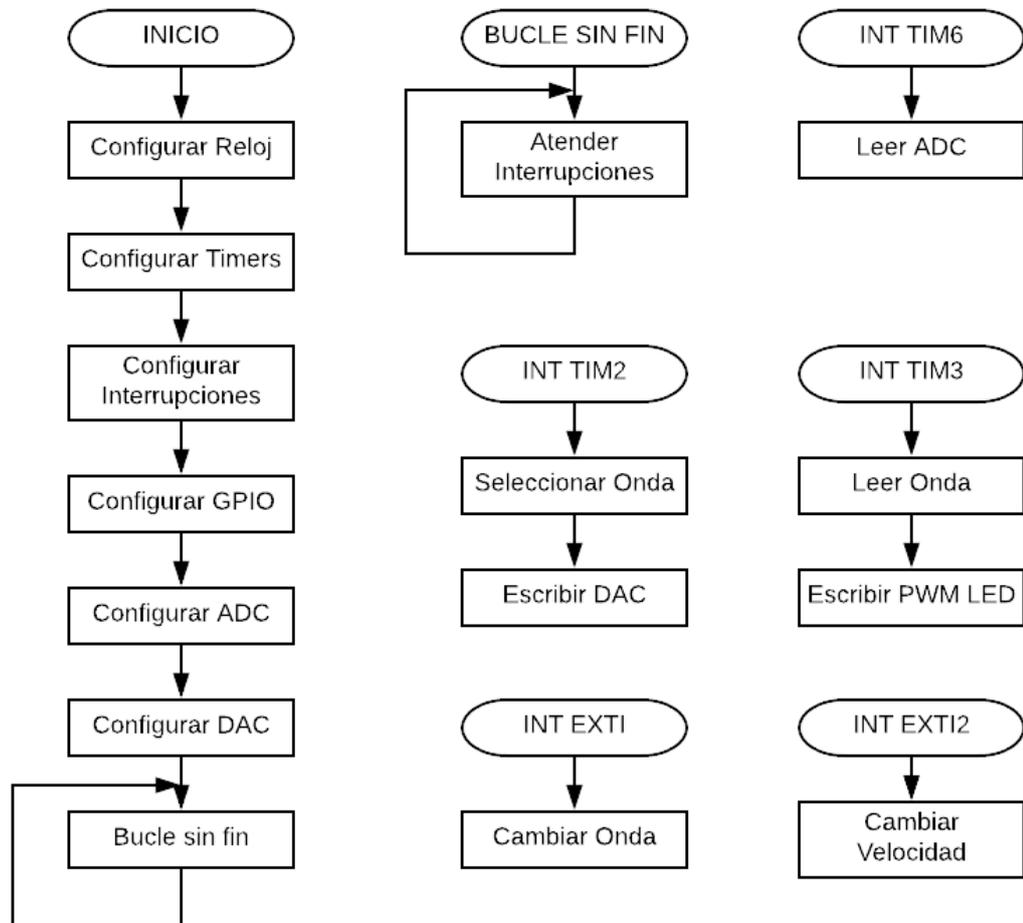
Diagrama de bloques del código:

Figura 12 - Diagrama de bloques del código del LFO.

Explicación del diagrama de bloques:

En la parte izquierda de la [Figura 12], se puede observar todo el proceso que sigue el microcontrolador desde que se inicia hasta que funciona en bucle sin fin. En cuanto al inicio, lo que hace el microcontrolador es **configurar y activar** los periféricos para asegurar que funcione de forma correcta el sistema, este paso solo se realiza una vez.

Cuando la configuración llega a su fin, se llega al **bucle sin fin**, esto quiere decir que el microcontrolador no saldrá de este estado a no ser que se reinicie o apague. En este estado sin fin es donde se desarrolla toda la actividad requerida del LFO:

- El bucle sin fin no recorre ninguna función, tan solo espera recibir las interrupciones para ejecutar código.
- Timer6 tiene una interrupción a 100Hz, y cuando salta procede a la lectura de las entradas analógicas y registrar su valor.

- Timer3 es la interrupción a 100Hz que gestiona la iluminación y color del led RGB de la placa.
- Timer2 es la interrupción que se encarga de manejar la frecuencia de la escritura en el DAC.
- Se atienden a las interrupciones externas de cambio de forma de onda y cambio de velocidad por separado.
- La interrupción externa del botón de velocidad ajusta el rango de frecuencia de la señal de salida, siendo 3 velocidades: baja, media y alta.
- La interrupción externa del botón de forma de onda está conectada a una entrada de jack de tipo trigger, así que cualquier evento en este botón o jack cambia la forma de onda de la señal de salida.

ADSR: Este módulo se emplea para dar expresividad a los sonidos, permitiendo ajustar la forma de un control de voltaje respecto al tiempo.

Esto hace que las señales de control de voltaje adquieran una forma más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo. Este ajuste tiene 4 partes progresivas:

Attack, Decay, Sustain, Release.

Para lograr este cometido, vamos a integrar el **ATMega328** en el sistema, ya que este microcontrolador tiene potencia suficiente para alcanzar los objetivos de este módulo. Para este dispositivo se necesita un gran número de **recursos del microcontrolador** para conseguir una operación correcta:

- **Interrupciones:** Necesarias para hacer una buena gestión de las lecturas de la entrada externa, evitan perder el tiempo leyendo en cada momento los estados de las entradas. Esta señal se encarga de activar el proceso de la envolvente y desactivarlo.
- **Entradas digitales:** Este módulo cuenta con tres entradas digitales, una asociada a la activación y desactivación de la envolvente, la segunda para el retrigger del evento de Attack, y la última sirve para seleccionar el rango de velocidad de la envolvente.
- **Entradas analógicas:** Necesarias para hacer una lectura de los valores de los potenciómetros para ajustar los tiempos de las 4 fases de la envolvente.
- **Salida analógica:** Su función es generar señal de control generada por los 4 pasos de la envolvente.

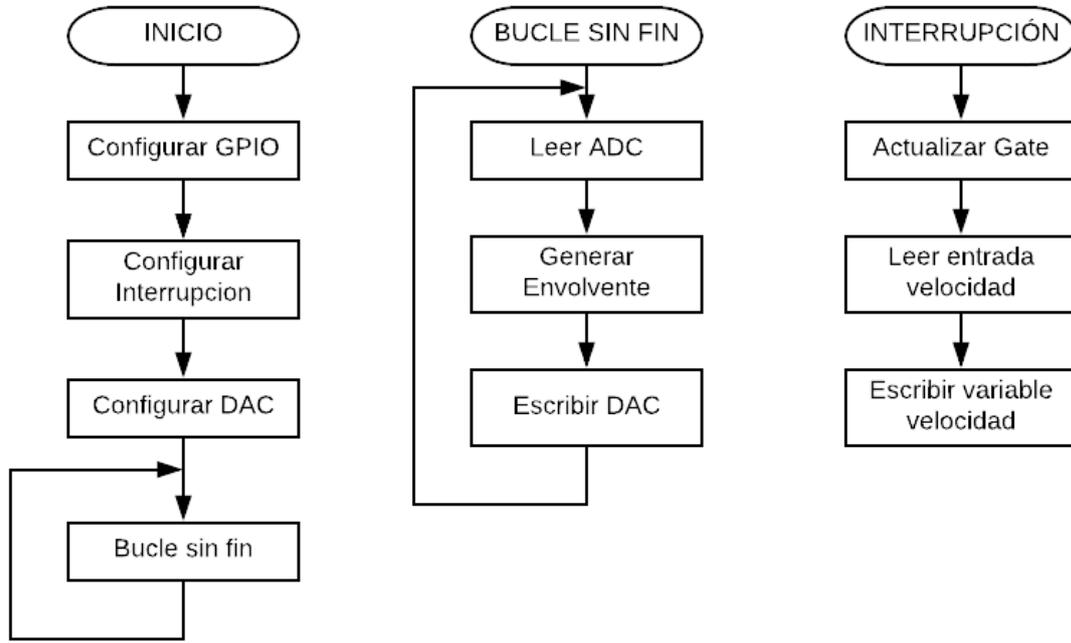
Diagrama de bloques del código:

Figura 13 - Diagrama de bloques del código del generador de Envolvente.

Explicación del diagrama de bloques:

En la parte izquierda de la [Figura 13], se puede observar todo el proceso que sigue el microcontrolador desde que se inicia hasta que funciona en bucle sin fin. En cuanto al inicio, lo que hace el microcontrolador es **configurar y activar** los periféricos para asegurar que funcione de forma correcta el sistema, este paso solo se realiza una vez.

Cuando la configuración llega a su fin, se llega al **bucle sin fin**, esto quiere decir que el microcontrolador no saldrá de este estado a no ser que se reinicie o apague. En este estado sin fin es donde se desarrolla toda la actividad requerida del generador de envolvente:

- Se leen las entradas analógicas para actualizar los tiempos de los diferentes estados del generador de envolvente.
- Se atiende a las interrupciones externas, utilizadas para indicar cambios en el procesamiento de la envolvente.

- La interrupción en el pin 2 sirve para indicar cambios en la señal de gate, ya que considera el flanco de subida y bajada, estos cambios son los que rigen la señal de salida y por tanto su procesado.
- La lectura del pin 4 sirve para indicar cambios en la señal de retrig, estos cambios son los que modifican el estado de la señal de salida cuando se activa..
- Se atiende procesado de la envolvente, encargados de actualizar la salida de señal de forma periódica y del control del led RGB.

8.4. Hardware:

De los 5 módulos que se compone este proyecto, se van a analizar 2 módulos de naturaleza digital, siendo el **VCO y LFO** ya que son los componentes del sistema más importantes y creados específicamente para este proyecto.

8.4.1. Diseño del esquema electrónico:

Para comprender el funcionamiento de los módulos, es útil mostrar el esquema electrónico para ver cómo se comunican los diferentes componentes del sistema.

VCO y LFO: Comparten una gran cantidad de componentes, por tanto se van a abordar como si fuesen el mismo hardware, haciendo distinciones cuando sea necesario.

- **Alimentación**

Para comenzar, lo más importante de este circuito es sin duda la alimentación, necesaria para un correcto funcionamiento. Los módulos se alimentan desde una fuente de alimentación que cumpla con el formato Eurorack [15]. Esto quiere decir que la fuente ofrece estos voltajes DC: **+12V**, **GND**, **-12V**. Estos voltajes se reciben en el conector presente en la imagen, y se añaden diodos para evitar problemas en caso de conectar al revés el conector.

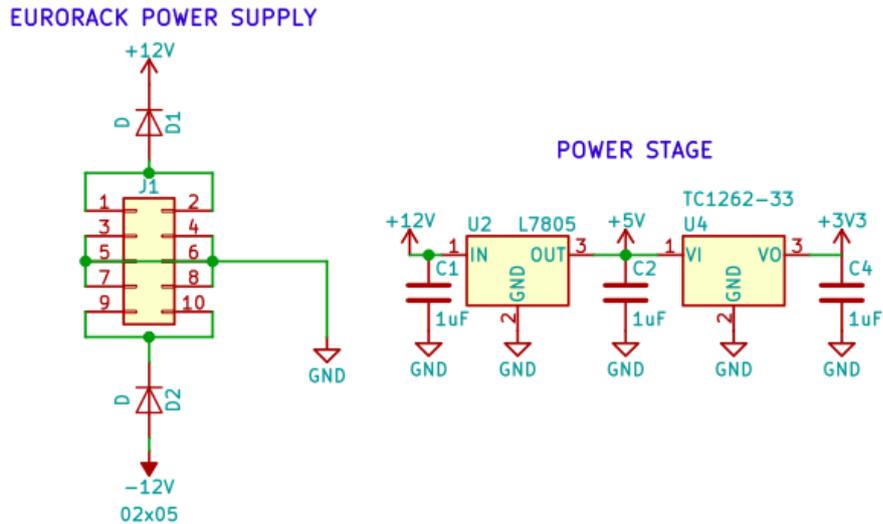


Figura 14 - Circuito de alimentación VCO / LFO.

En la parte derecha de la [Figura 14], se observan los 2 circuitos integrados encargados de proveer de **5V** (L7805) y **3.3V** (TC1262-33) al circuito, cuentan ambos con condensadores de desacoplo para asegurar un funcionamiento estable.

- **Desacoplo de alimentación**

Circuito empleado para garantizar la estabilidad de los voltajes que provienen de la fuente de alimentación, así como para eliminar los posibles transitorios presentes en ella, se muestra en la [Figura 15].

Cabe destacar el diferenciamiento entre los planos de alimentación analógico y digital, para evitar indeseadas interferencias entre ellos.

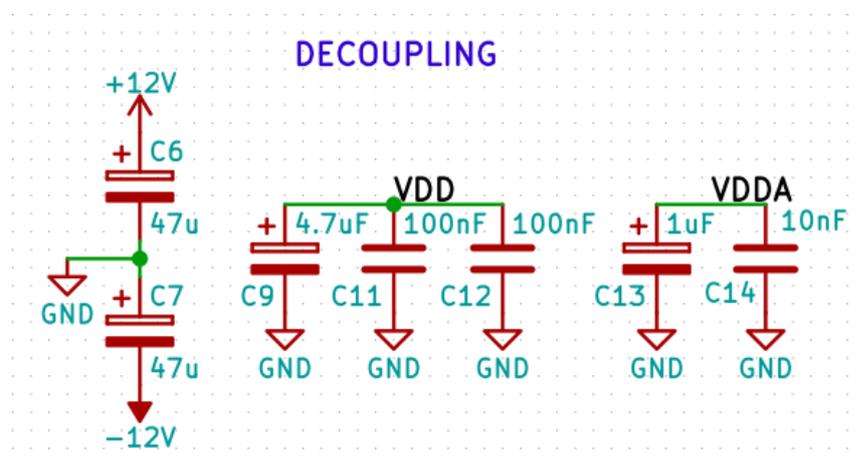


Figura 15 - Circuito de desacoplo de alimentación VCO / LFO.

- **Conector SWD**

Conector utilizado para la programación y depuración del dispositivo mediante un dispositivo STLink V2.

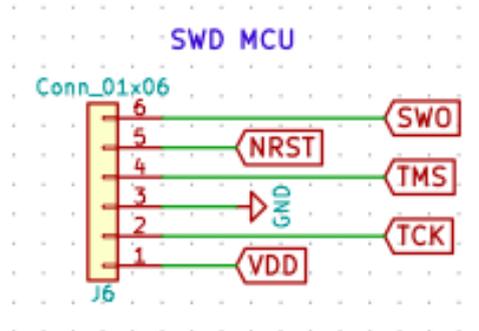


Figura 16 - Circuito del conector SWD del VCO / LFO.

- **Entrada de botón y jack forma de onda**

Circuito utilizado para la selección de la forma de onda, tanto con un botón integrado en el módulo, como con una entrada de jack externa.

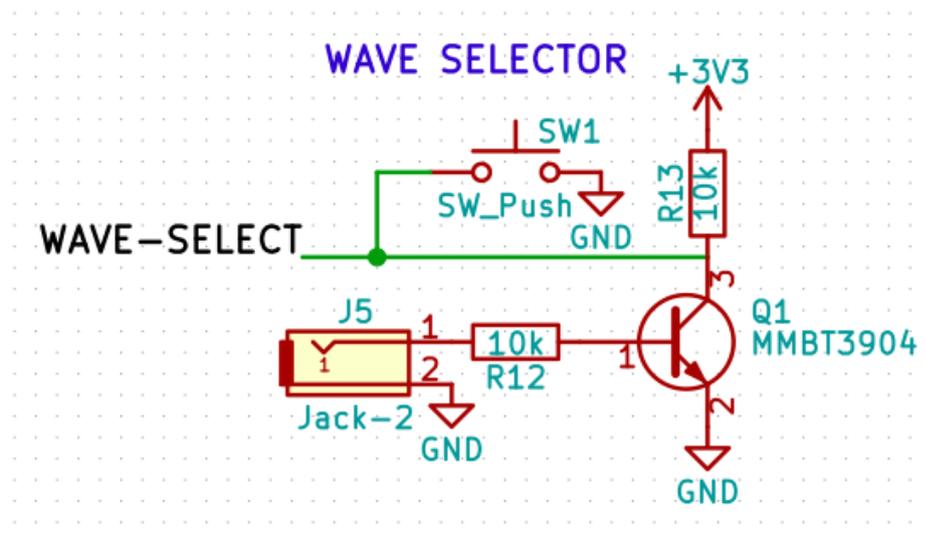


Figura 17 - Circuito de entrada de selector de onda VCO / LFO.

- **Led RGB**

Circuito utilizado para el funcionamiento del led RGB integrado en el módulo.

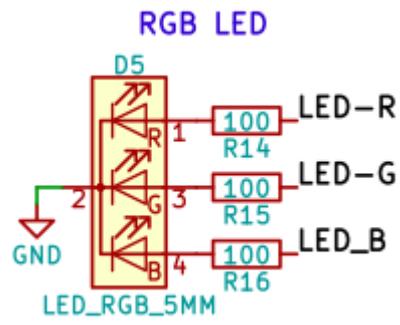


Figura 18 - Circuito led RGB del VCO / LFO.

Circuitos exclusivos para el VCO:

- **Entradas analógicas VCO**

Circuito utilizado para el ajuste de la frecuencia de la señal de salida, permite ajustes fino y grueso, además de contar con una entrada para jack externo con atenuador. Esta entrada está limitada por el diodo Zener a 3V3.

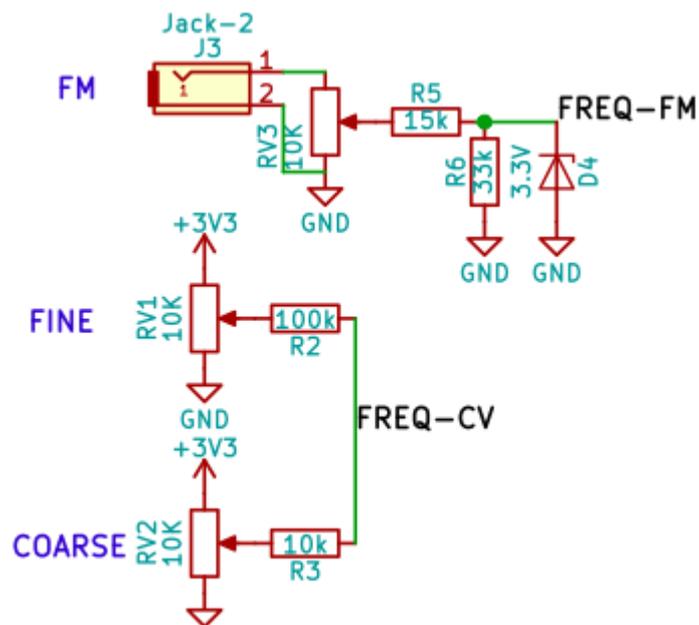


Figura 19 - Circuito de entradas analógicas del VCO

- **Entrada 1 Voltio / Octava VCO**

Circuito utilizado para el ajuste de la frecuencia en el protocolo 1 voltio por octava, esta entrada es atenuada con una ganancia equivalente a $G = -0.68$, para después pasar por otro operacional con ganancia equivalente a $G = -1$, que integra un filtro paso bajo de frecuencia de corte 16kHz, dejando la señal en un rango perfectamente legible para el ADC, mostrado en la [Figura 20].

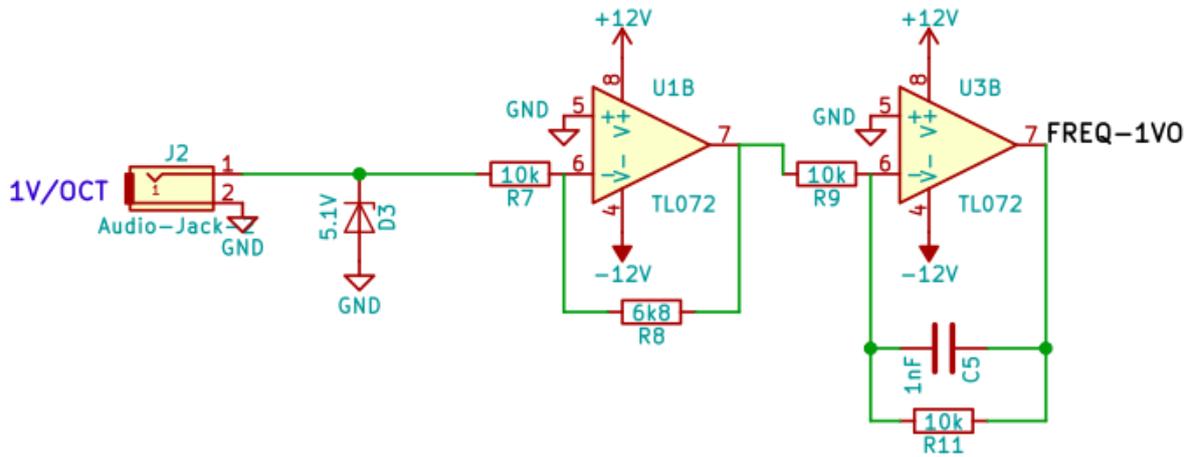
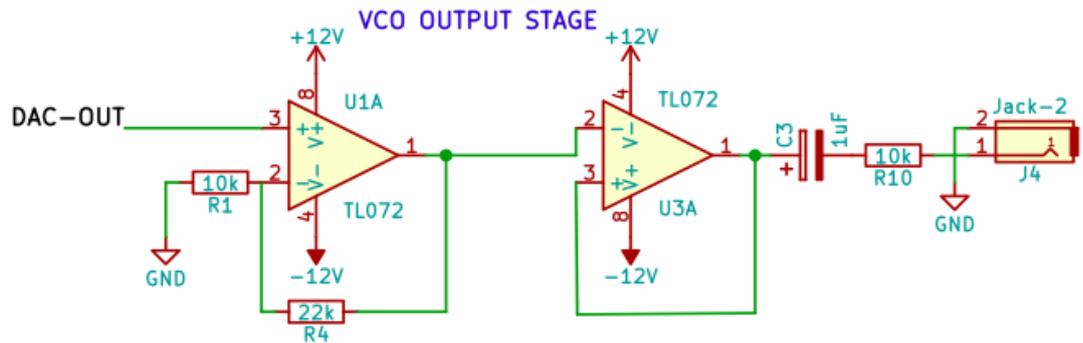


Figura 20 - Circuito de entrada 1 Voltio / Octava del VCO

- **Salida de señal VCO**

Circuito utilizado para la señal de salida, primero se aumenta su amplitud 3.2 veces, y después se pasa la señal por un filtro paso alto, eliminando el componente DC, por tanto sale una señal de amplitud $\pm 5V$.



F
Figura 21 - Circuito de salida de señal del VCO

- **Microcontrolador VCO**

Circuito realizado para el microcontrolador, el corazón del módulo, recibe todas las entradas y salidas, así como el conector SWD, dispone de alimentación a 3V3 y la señal de reloj proviene del oscilador a 8MHz.

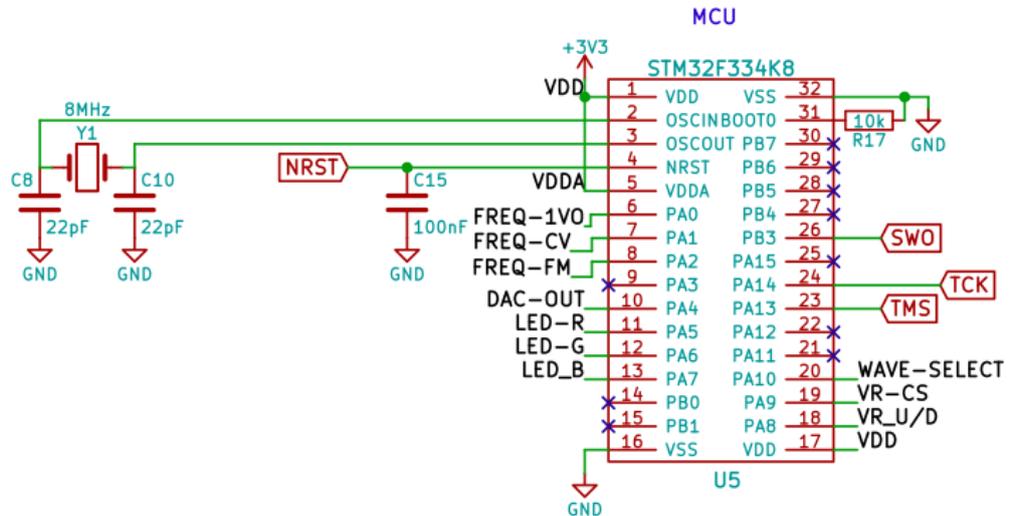


Figura 22 - Circuito del microcontrolador del VCO

Circuitos exclusivos para el LFO:

- **Entrada de botón de velocidad LFO**

Circuito utilizado para la selección del rango de velocidad, con un botón integrado en el módulo.



Figura 23 - Circuito botón de velocidad del LFO.

- **Microcontrolador LFO**

Circuito realizado para el microcontrolador, el corazón del módulo, recibe todas las entradas y salidas, así como el conector SWD, dispone de alimentación a 3V3 y la señal de reloj proviene del oscilador a 8MHz.

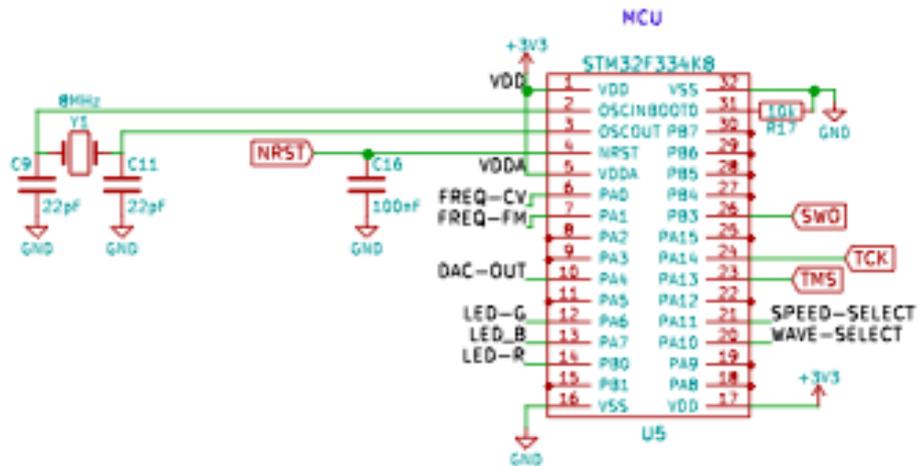


Figura 24 - Circuito del microcontrolador del LFO.

ADSR: Este módulo es un complemento para el sistema de síntesis sustractivo, por tanto, se muestran sus circuitos en este apartado sin una profundización exhaustiva.

- **Circuito de alimentación y desacoplo**

Se coge la alimentación del conector de la fuente de Eurorack con diodos de protección para evitar voltajes inversos a los requeridos. También incluye un regulador lineal de 5V para el circuito del microcontrolador, y condensadores de desacoplo para la alimentación del circuito.

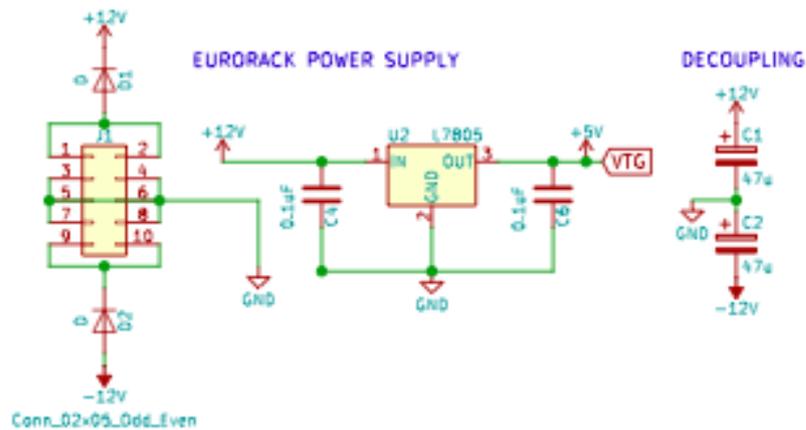
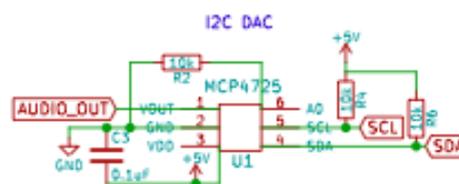


Figura 25 - Circuito de alimentación y desacoplo del ADSR

- **Circuito de conversor digital-analógico**

Circuito empleado para generar la salida analógica del módulo, es controlada



mediante el protocolo i2c a través del microcontrolador.

Figura 26 - Conversor analógico-digital del ADSR

- **Salida de señal**

Circuito empleado para generar la salida analógica del módulo ya amplificada y pasada a través de un buffer, permite elegir entre una salida de [0,5]V o [0,8]V. También se incluye un led controlado por PWM para visualizar el nivel de la señal.

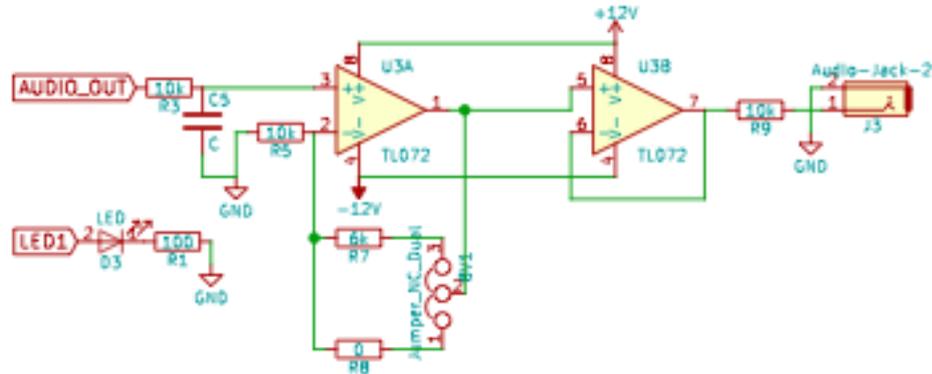


Figura 27 - Salida de señal del ADSR

- **Conector ISP**

Conector utilizado para la programación y depuración del dispositivo mediante un dispositivo Arduino.

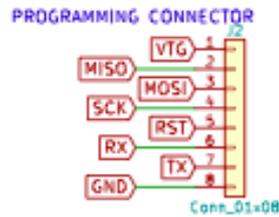


Figura 28 - Conector ISP del ADSR

- **Microcontrolador ADSR**

Circuito realizado para el microcontrolador, el corazón del módulo, recibe todas las entradas y salidas, así como el conector ISP, dispone de alimentación a 5V y la señal de reloj proviene del oscilador a 16MHz.

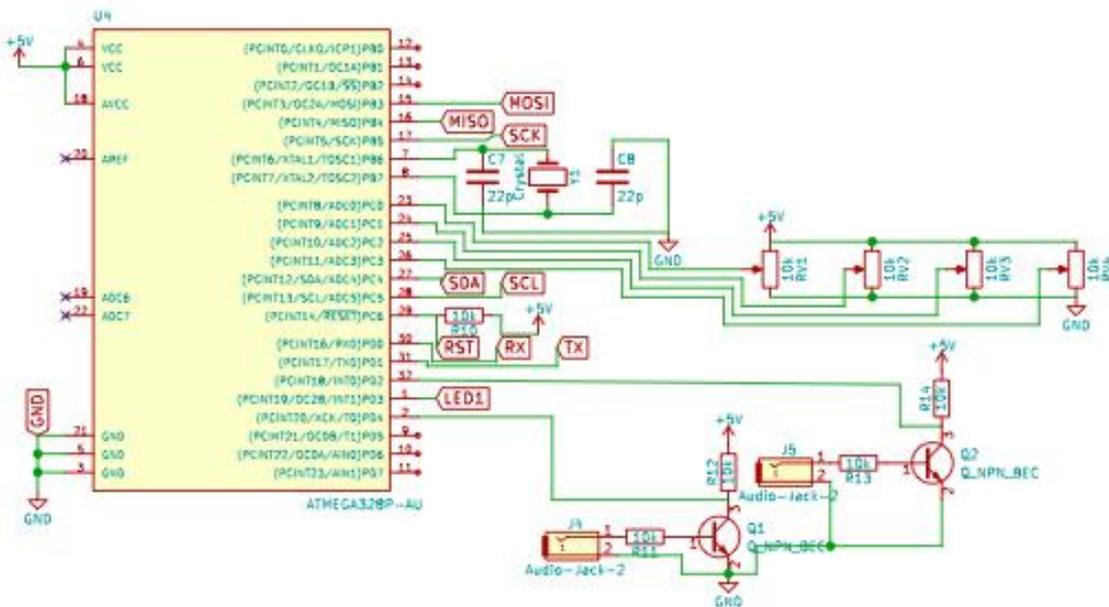


Figura 29 - Microcontrolador del ADSR

VCF: Este módulo es un complemento para el sistema de síntesis sustractivo, por tanto se muestran sus circuitos en este apartado sin una profundización exhaustiva.

- **Circuito de alimentación y desacoplo**

Se coge la alimentación del conector de la fuente de Eurorack con diodos de protección para evitar voltajes inversos a los requeridos. También incluye una referencia de -5V para el circuito de CV frecuencia, y condensadores de desacoplo para la alimentación del circuito.

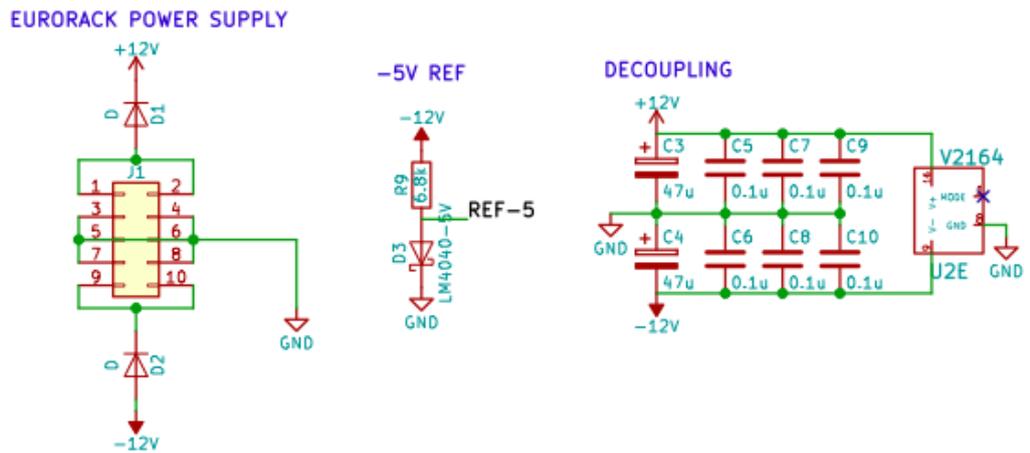


Figura 30 - Circuito de alimentación y desacoplo del VCF.

- **Circuito de CV de la frecuencia de corte**

Dispone de una entrada de CV con atenuador e inversor integrado, otra entrada de 1V/Octava, y el potenciómetro para ajustar esta frecuencia de corte.

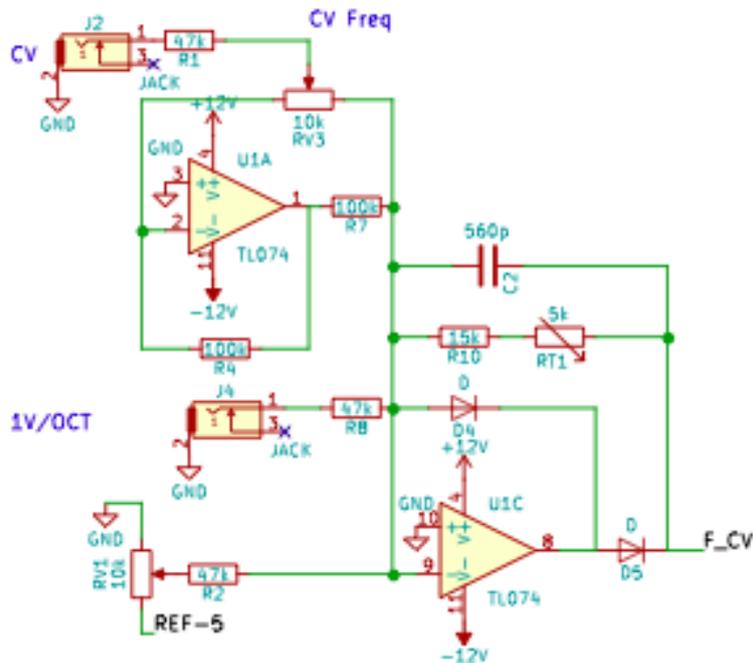


Figura 31 - Circuito del control de frecuencia del VCF.

- **Circuito de CV de la resonancia**

Dispone de una entrada de CV con atenuador integrado, y el potenciómetro para ajustar esta resonancia.

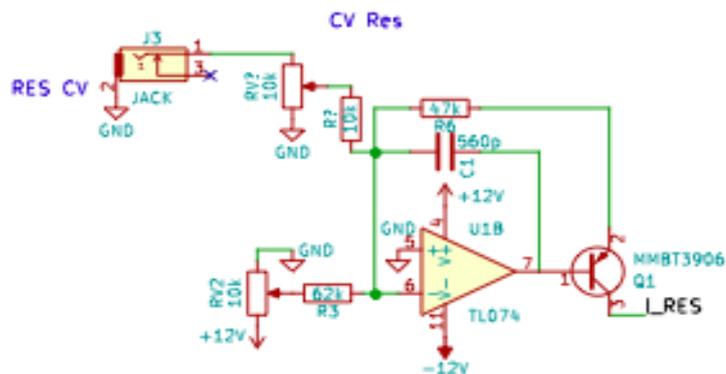


Figura 32 - Circuito del control de resonancia del VCF.

- **Circuito del filtro**

Este es el corazón del módulo, se encarga de filtrar la señal entrante con la modulación de las señales de CV.

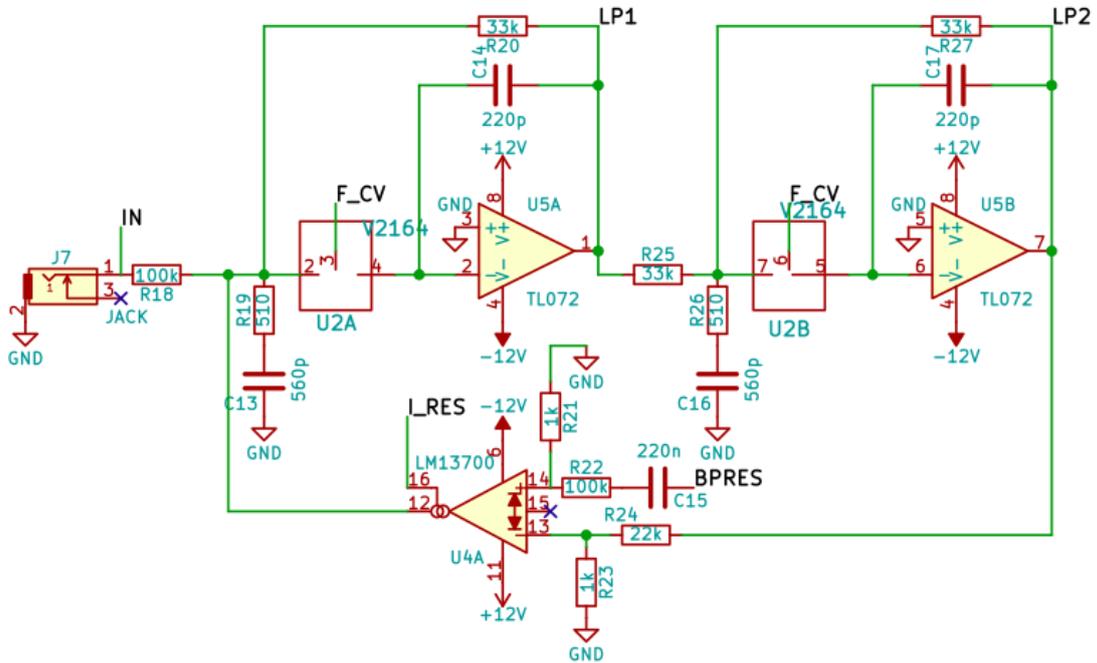


Figura 33 - Circuito del filtro del VCF.

- **Salidas del filtro**

Salidas del filtro, incluye una salida paso bajo y otra paso banda.

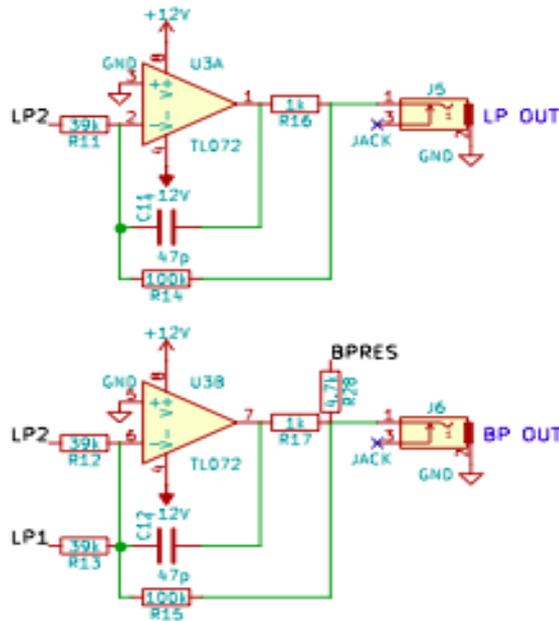


Figura 34 - Circuito de salidas del VCF.

VCA: Este módulo es un complemento para el sistema de síntesis sustractivo, por tanto se muestran sus circuitos en este apartado sin una profundización exhaustiva.

- **Circuito de alimentación y desacoplo**

Se coge la alimentación del conector de la fuente de Eurorack con diodos de protección para evitar voltajes inversos a los requeridos y condensadores de desacoplo para la alimentación del circuito.

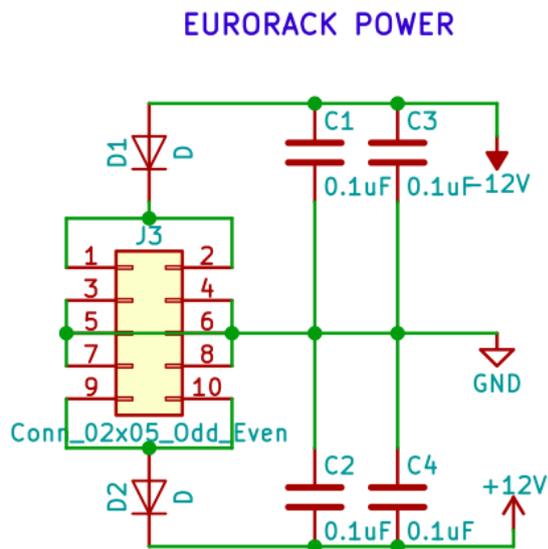


Figura 35 - Circuito de alimentación y desacoplo del VCA.

- **Circuito de CV de la ganancia**

Dispone de una entrada de CV con atenuador integrado, otra sin atenuador y el potenciómetro para ajustar esta ganancia así como un led para visualizarla.

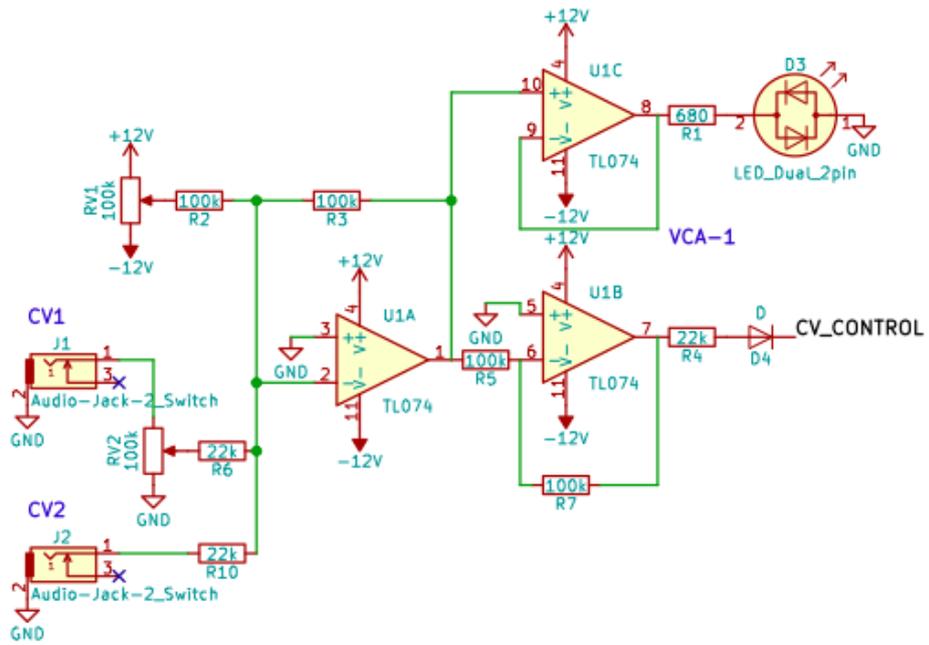


Figura 36 - Circuito del control de la ganancia del VCA.

- **Circuito de VCA**

Dispone de una entrada de audio, después un amplificador de transconductancia que varía la ganancia dependiendo del circuito de control del VCA.

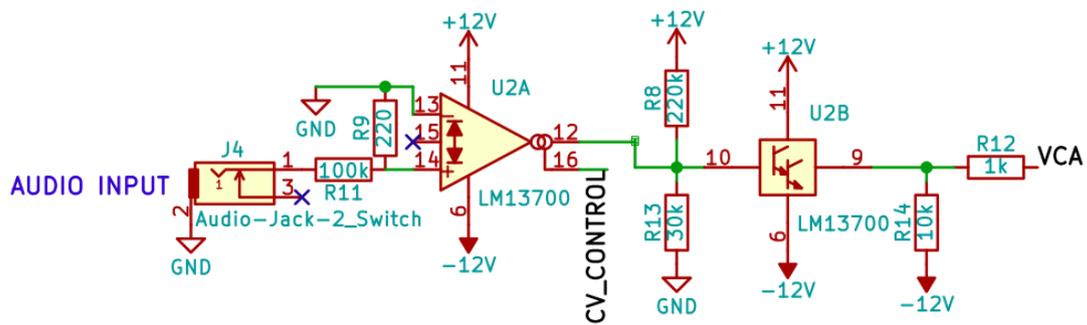


Figura 37 - Circuito del amplificador del VCA.

- **Circuito de salida**

Dispone de una salida de audio, esta señal es filtrada con un paso alto a 16Hz para posteriormente pasar por un buffer.

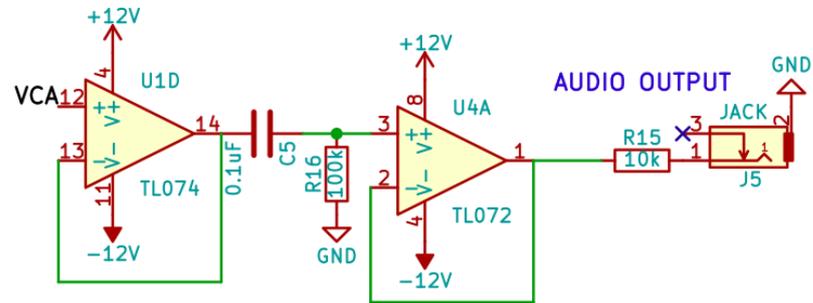


Figura 38 - Circuito de salida del VCA.

8.4.2. Diseño de la PCB:

La implementación del circuito impreso es un paso muy importante, ya que es decisivo poder trazar el circuito en un espacio reducido, optimizando al máximo los recursos, para poder integrar el sistema en espacios reducidos.

Las características recogidas en este apartado son comunes para los módulos de este sistema:

VCO, LFO, VCA, VCF Y ADSR:

Por estos requerimientos de espacio, se ha decidido usar componentes de montaje superficial (**SMD**) [27]

- Resistencias y condensadores, estos componentes son de la medida **1206**.
- Diodos son de la medida **SOD-123**.
- Potenciómetros, conectores jack, tiras de pines, cristal resonador, botones y LEDs son de **agujero pasante**.
- Todos los circuitos integrados son variaciones de **SMD** (SOIC-16, TO-252, LQFP32...).

En cuanto el esquema electrónico está diseñado, y a cada componente se le ha asignado un empaquetado específico, KiCad es capaz de generar una **lista de redes** [28], que servirá para trazar la PCB sin errores, ya que se sigue el esquema trazado anteriormente.

Por tanto, con los componentes en su empaquetado correspondiente, y la lista de redes preparada, podemos pasar a diseñar la PCB.

Dimensiones: Nos vamos a ajustar a las medidas **100x30mm**, ya que 30mm será el grosor final del módulo, y 100mm es la máxima medida permitida en los fabricantes de PCBs para realizar una oferta por ser una placa pequeña. Una vez están delimitados los bordes, tenemos que empezar a poblar la placa con los componentes.

Componentes: Debido a la densidad de componentes en esta placa, vamos a ocupar las 2 capas del circuito impreso. Por tanto habrán componentes distribuidos por las dos capas, tanto SMD como de agujero pasante.

Pistas: La alta densidad de conexiones requeridas y componentes en esta placa hacen que sea necesario diseñar las pistas con un grosor de **0.2mm**, una medida relativamente pequeña pero que no dará problemas.

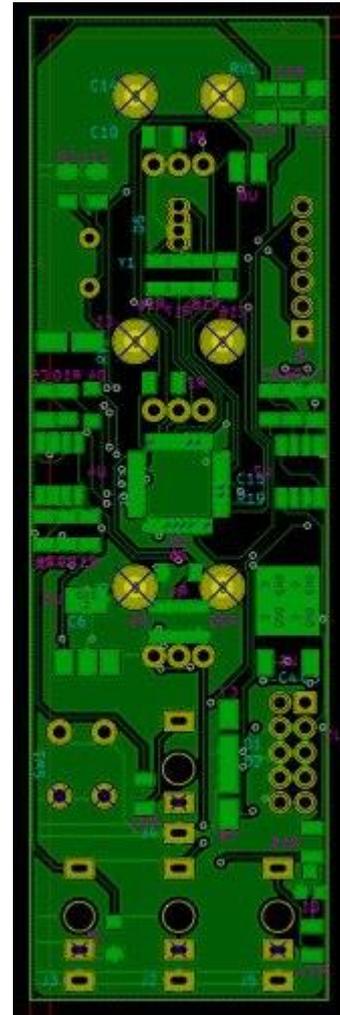
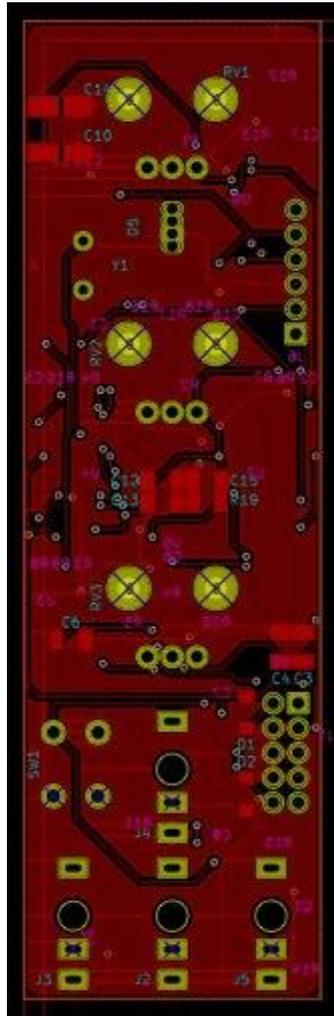


Figura 39 - Parte superior PCB del VCO Figura 40 - Parte inferior PCB del VCO

Una vez diseñada la PCB, se procede a comprobar que no hay ningún defecto en el diseño, como podría ser una pista demasiado cerca de otra, para ello se emplea la herramienta de KiCad **DRC Check** [29] ya que avisa en caso de que encuentre algún problema.

En el caso de que no haya ningún problema, ya se pueden exportar estos archivos en formato **Gerber** [30] para así enviarlos al fabricante que se vaya a encargar de fabricar la PCB.

8.5. Componentes mecánicos:

Con la PCB ya diseñada y fabricada, hay que componer el panel del módulo para que sujete todos los componentes mecánicos como potenciómetros y conectores jack.

8.5.1. Diseño de los paneles:

Las características recogidas en este apartado son comunes para los módulos de este sistema:

VCO, LFO, VCA, VCF Y ADSR:

Para el panel que sujeta los componentes mecánicos, se ha optado por realizar un diseño simple, que deje claras las entradas, salidas y ajustes disponibles en este módulo. Por tanto se va a proceder a indicar qué característica modifica cada potenciómetro, también información sobre las entradas y salidas.

También se le ha incluido en la esquina de abajo derecha, la versión simplificada del logo de nuestra marca.

Este panel será cortado a láser con una máquina CNC. Esta máquina realizará dos operaciones:

- **Corte:** Para separar los bordes y agujeros por los que se pasarán los potenciómetros botones, LEDs y conectores Jack.
- **Grabado:** Para realizar las marcas e inscripciones en el módulo, que aclara la función de cada potenciómetro, botón, entrada y salida.

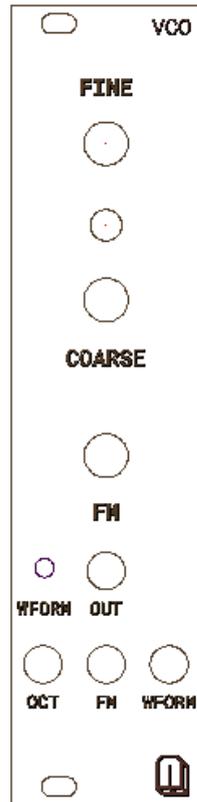


Figura 41 - Diseño del panel del VCO



Figura 42 - Diseño del panel del VCF

9. Implementación:

Una vez se ha comentado el diseño del dispositivo, se procede a indicar la implementación que ha seguido, es decir, el orden lógico de los procesos que componen este sistema de síntesis de audio. Para esta sección se ha detallado la implementación del módulo VCO, ya que en este módulo se pueden ver todas las fases del desarrollo. Para aclarar la implementación de los diferentes módulos, se detalla su implementación aquí:

- **VCO:** Prototipado en placa board, diseño del hardware, diseño del software con el entorno Keil y las librerías de NANO, y diseño del panel.
- **LFO:** Prototipado en placa board, diseño del hardware, diseño del software con el entorno Keil y las librerías de NANO, y diseño del panel.
- **ADSR:** Prototipado en placa board, diseño del hardware, diseño del software con el entorno Arduino y las librerías externas, y diseño del panel.
- **VCF:** Prototipado en placa board, diseño del hardware, y diseño del panel.
- **VCA:** Prototipado en placa board, diseño del hardware, y diseño del panel.

9.1. Prototipado en placa board:

Para comenzar con el diseño del módulo de VCO del sistema, se ha comprado una placa de desarrollo del microcontrolador que se va a utilizar, **STM32F334**. Esta placa forma parte de la gama llamada **NUCLEO** de STM [31]. Concretamente vamos a utilizar el modelo NUCLEO-F334R8, ya que es el más ajustado a nuestras necesidades.

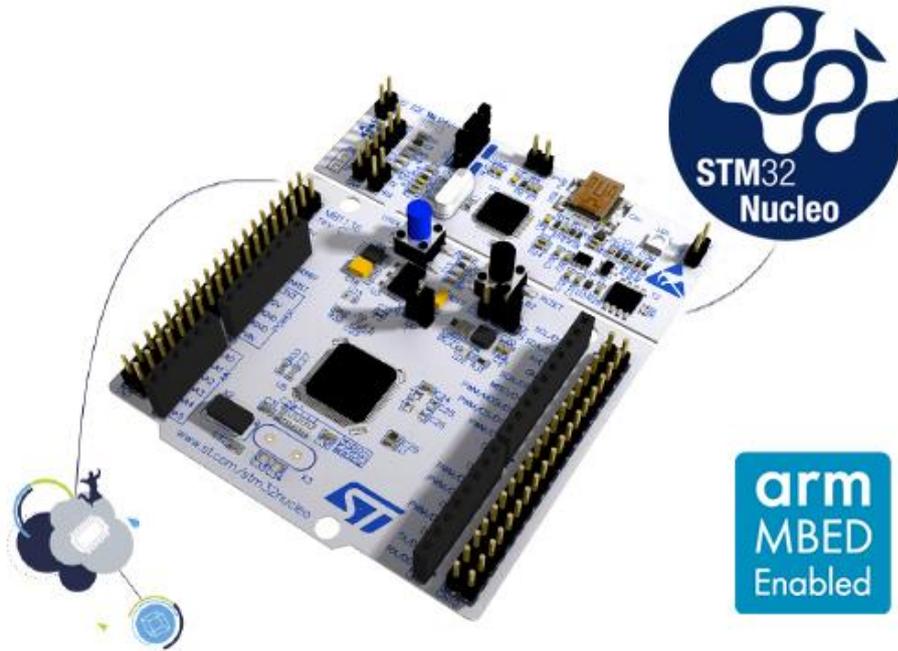


Figura 43 - Imagen de la placa de desarrollo NUCLEO.

Con esta placa de desarrollo podremos prototipar algunas funciones del VCO, así como comprobar su funcionamiento y empezar a escribir algunas líneas de código. Para complementar a esta placa de desarrollo, se va a incluir una pequeña placa board para alojar componentes de prueba como potenciómetros, LEDs y salidas jack.

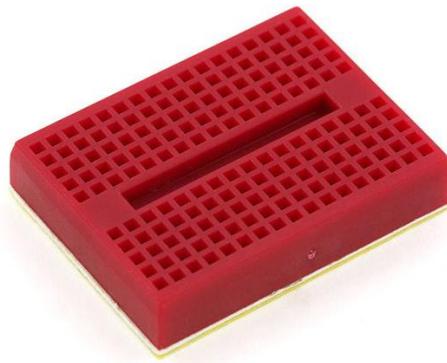


Figura 44 - Imagen de la placa board utilizada para el prototipo.

Esta configuración se queda corta en seguida, ya que no se puede integrar hardware con facilidad, por tanto una vez concluido el test del microcontrolador y las entradas y salidas, se decide dar el salto a la fabricación de un circuito específico para este sistema.

9.2. Diseño del hardware:

En primer lugar, se debe trazar un esquemático con los componentes y circuitos requeridos en el sistema. A este esquemático se le asigna unos componentes con sus respectivos atributos mecánicos. Con los componentes, sus huellas y el circuito ya seleccionado, se crea una lista de redes y se traza la PCB en función de esta. Esta secuencia de pasos se ha explicado detalladamente en el apartado anterior.

La función de este hardware es poder probar todas las entradas y salidas, así como una operación normal del dispositivo ya incluyendo todos los componentes necesarios para su funcionamiento.

Una vez diseñada la PCB, hay que exportar los archivos Gerber y enviarlos al fabricante para su producción.

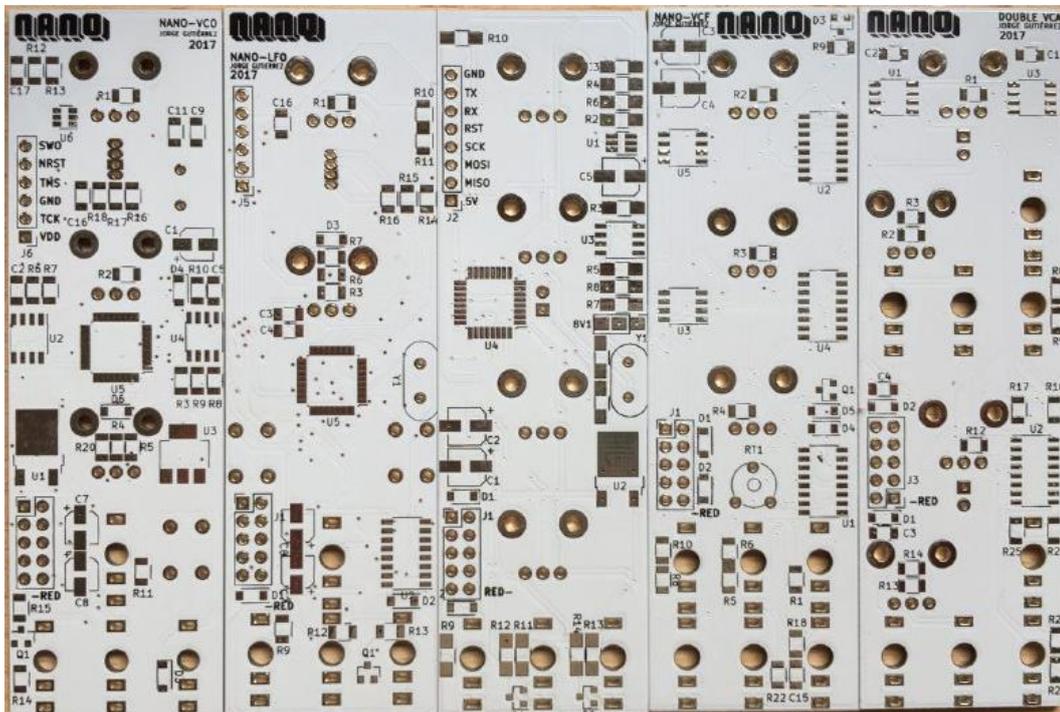


Figura 45 - PCBs de los 5 módulos del proyecto

Una vez recibida la PCB, se debe proceder a su montaje, que llevará varias horas al tener tantos componentes. Primero se sueldan los componentes SMD, los más pequeños y después se pasan a componentes de agujero pasante. Antes de conectar el módulo a la corriente, se debe verificar que todas las soldaduras están bien realizadas y los componentes bien colocados.

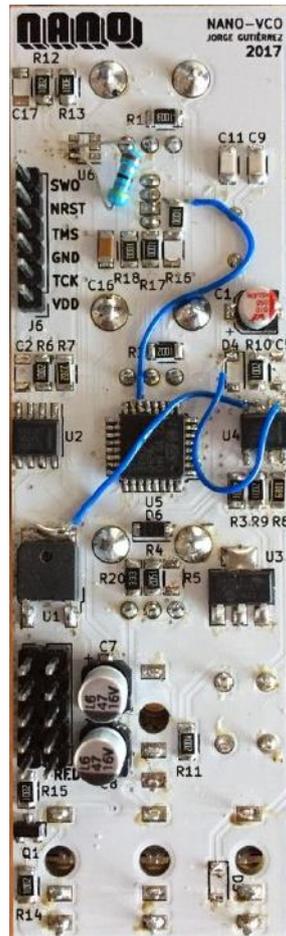


Figura 46 - Imagen de la PCB del VCO con los componentes.

Una vez comprobados los componentes, se conecta el módulo a la corriente, si no da ningún cortocircuito, podemos conectar el programador STLink V2 e intentar subir un código de prueba, si este código de prueba funcionase, se dispondría de un módulo hardware ya preparado para empezar a programar.

9.3. Diseño del software:

Con el dispositivo hardware ya en funcionamiento, el primer paso en cuanto al software es crear librerías del proyecto para facilitar la escritura del código, esto es un proceso costoso pero que si se realiza de la forma correcta, solo se tendrá que hacer una vez, mientras que se utilizará muchas.

- **Adc_nano.h:** Librería escrita para el manejo del conversor analógico digital (ADC), su inicialización y procesado.
- **Dac_nano.h:** Librería escrita para el manejo del conversor digital anaógico (DAC), su inicialización y procesado.
- **Dma_nano.h:** Librería escrita para el manejo del acceso directo a memoria (DMA), su inicialización y procesado.
- **Exti_nano.h:** Librería escrita para el manejo de las interrupciones externas (EXTI), su inicialización y procesado.
- **Gpio_nano.h:** Librería escrita para el manejo de las entradas y salidas (GPIO), su inicialización y procesado.
- **Pwm_nano.h:** Librería escrita para el manejo del módulo PWM, su inicialización y procesado.
- **Tim_nano.h:** Librería escrita para el manejo de los timers internos del microcontrolador (TIM), su inicialización y procesado.

Una vez que las librerías están creadas, se puede comenzar a escribir el código del VCO con la ayuda de estas.

Diagrama de bloques del código:

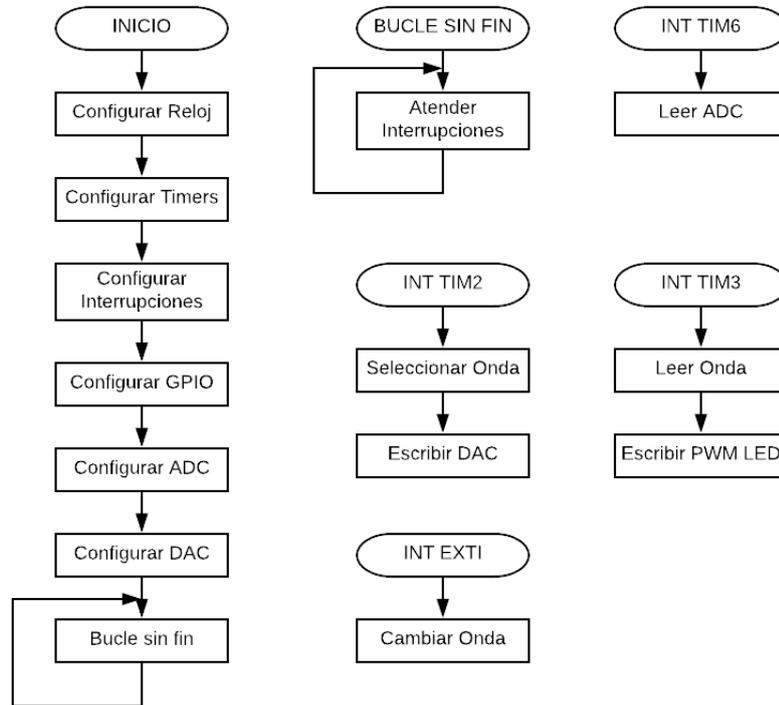


Figura 47 - Diagrama de bloques del código del VCO.

Este diagrama de bloques ya ha sido explicado anteriormente en el apartado de **diseño de software**, así como todos los procesos que sigue el código del microcontrolador. Por tanto, tenemos todo el código necesario para poner en marcha el VCO y testear su funcionamiento.

9.4. Diseño del panel:

Se procede a diseñar el panel que sujeta los componentes mecánicos, este panel será cortado a láser con una máquina CNC. Esta máquina realizará la operación de corte y grabado, que le dará el diseño final a esta pieza.

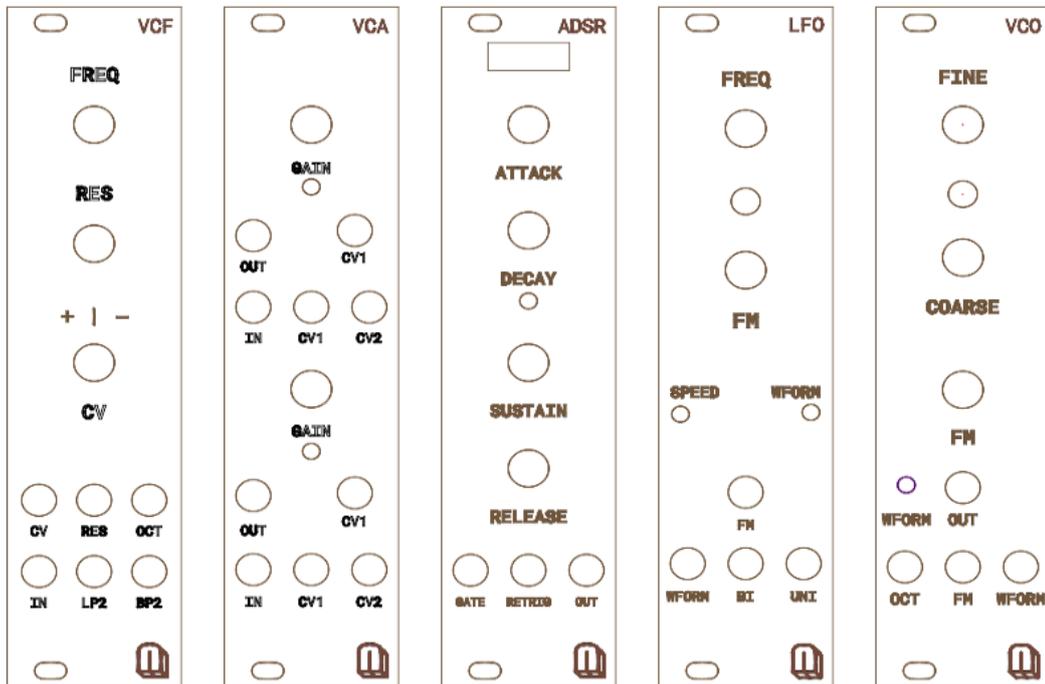


Figura 48 - Diseño de los paneles de los 5 módulos

Una vez acabado el panel, ya podemos montarlo al módulo e incluirlo en el sistema de síntesis sustractiva, ya que ahora dispone de las características necesarias para montarlo en el formato Eurorack.

10. Validación del dispositivo:

10.1. Introducción

Este apartado es quizá el más importante, ya que se procede a analizar el comportamiento de los módulos y medir si cumplen con las especificaciones planteadas en la definición de requisitos. Para comprobar estas funcionalidades, se van a realizar varios tipos de mediciones para corroborar su cumplimiento.

10.2. Testeo del hardware

Al disponer de dos tecnologías diferentes, se va a proceder a testear el hardware de un forma diferente:

Módulos analógicos: En estos módulos, el hardware es el que cumple con las especificaciones requeridas, por tanto el testeo consiste en primero una prueba de encendido, seguido de una prueba para ver si los componentes no superan temperaturas normales, y ya se pasa a comprobar las funcionalidades del módulo.

- **Prueba de encendido:** El primer paso después del montaje de los módulos analógicos es conectarlo a la corriente y verificar si puede encenderse sin problemas. Esta prueba descarta cortocircuitos y errores severos en el montaje o diseño. En caso de conseguir un encendido correcto, sin problemas, se procede a la siguiente prueba, en el caso contrario, hay que buscar un problema grave en el dispositivo.
- **Prueba de calentamiento:** Una vez encendido el módulo, es importante comprobar que las temperaturas de los componentes sean correctas, ya que una temperatura demasiado elevada nos indica errores de hardware o cortocircuitos. Para realizar esta prueba se debe encender el módulo y monitorizar sus temperaturas durante unos minutos, si no se salen de lo común, se procede a la siguiente prueba, en el caso contrario, hay que buscar un problema en el dispositivo.
- **Prueba de funcionalidades:** En este apartado se comprueban las especificaciones planteadas respecto a los módulos analógicos al inicio del documento.

VCA: Amplifica o atenúa la señal con control de voltaje, ajustando la ganancia requerida. En este apartado se emplea la **señal amarilla como referencia** de señal de entrada, y la **azul como salida**.

En la figura 49 podemos observar la señal, que en este caso es nula al estar el potenciómetro de ajuste de ganancia a 0, y no cuenta con señal de control.

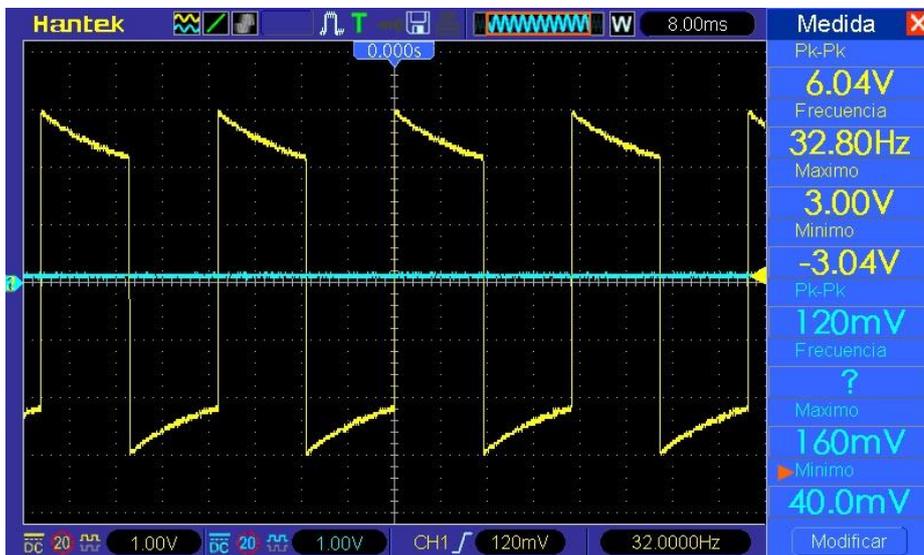


Figura 49 - VCA completamente cerrado

En la figura 50 podemos observar la señal, que en este caso es aproximadamente la mitad de la entrada al estar el potenciómetro de ajuste de ganancia a la mitad, y no cuenta con señal de control.

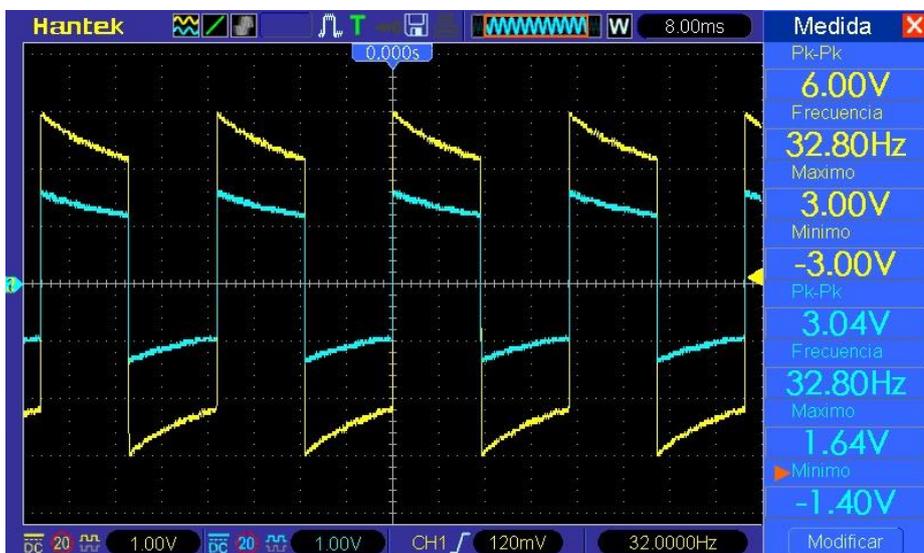


Figura 50 - VCA control a la mitad

En la figura 51 podemos observar la señal, que en este caso es superior a la entrada al estar el potenciómetro de ajuste de ganancia al máximo, y no cuenta con señal de control.

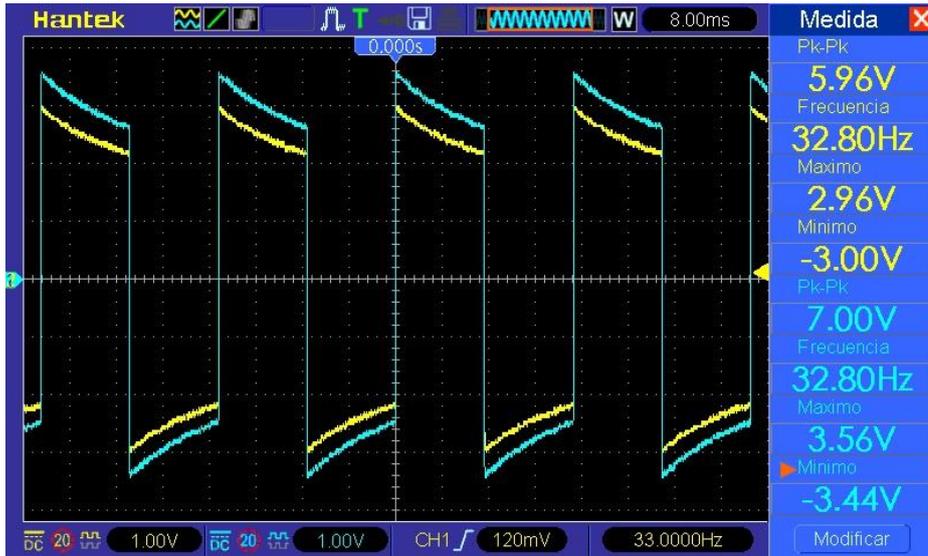


Figura 51 - VCA completamente abierto

En la figura 52 podemos observar la señal, que en este caso es variante con un patrón sinusoidal al estar el potenciómetro de ajuste de ganancia a 0, y cuenta con señal de control sinusoidal que ajusta la ganancia.

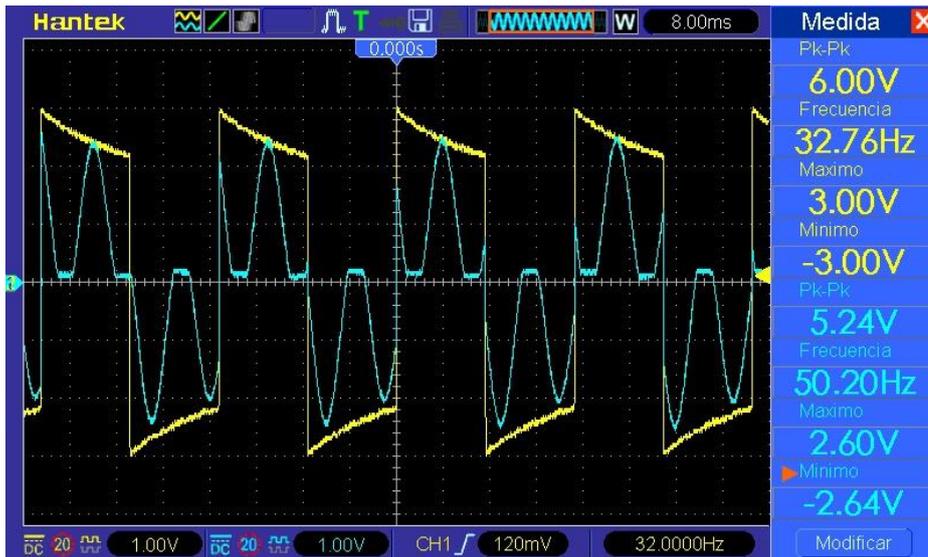


Figura 52 - VCA modulado por una senoidal

En la figura 53 podemos observar la señal, que en este caso es variante con un patrón sinusoidal al estar el potenciómetro de ajuste de ganancia a la mitad, y cuenta con señal de control sinusoidal que ajusta la ganancia.

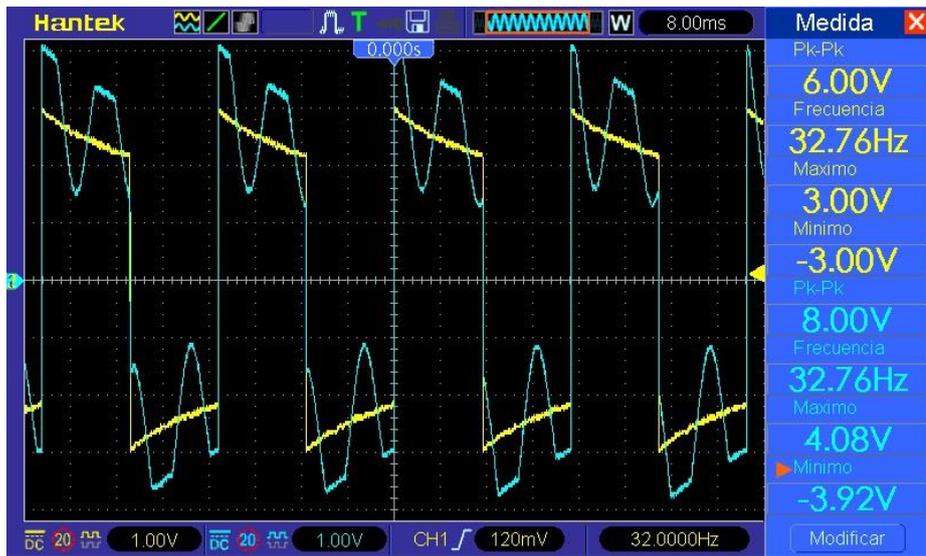


Figura 53 - VCA modulado por una senoidal y potenciómetro a la mitad

VCF: Filtra la señal de audio en torno a frecuencias comprendidas entre [20-20000]Hz, pudiendo ajustar la resonancia y frecuencia de corte del filtro y tener dos salidas simultáneas, Low Pass y Band Pass. En este apartado se emplea la **señal azul como referencia** de señal de entrada, y la **amarilla como salida**.

En la figura 54 y 55 podemos observar la señal filtrada estar el potenciómetro de ajuste de frecuencia de corte a 0.

- En la salida de Low Pass la señal tan solo sale con los armónicos más bajos.
- En la salida de Band Pass la señal tan solo sale con los armónicos más bajos.

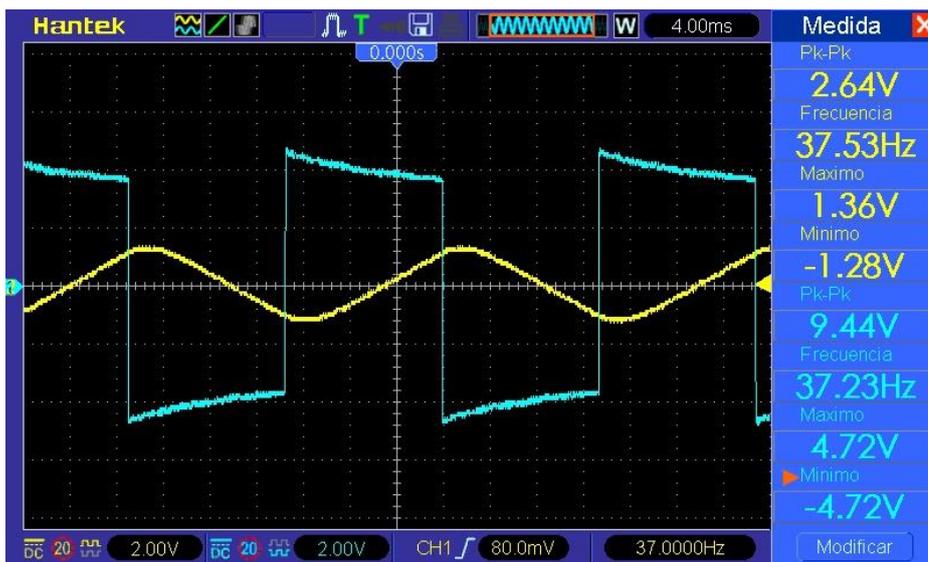


Figura 54 - VCF con frecuencia de corte a 0, salida LP

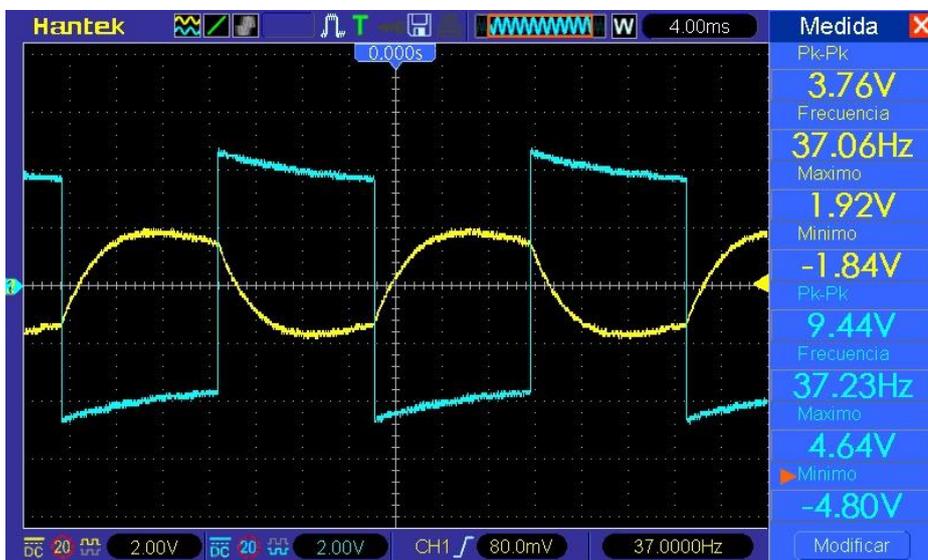


Figura 55 - VCF con frecuencia de corte a 0, salida BP

En la figura 56 y 57 podemos observar la señal filtrada estar el potenciómetro de ajuste de frecuencia de corte a la mitad.

- En la salida de Low Pass la señal sale con contenidos armónicos bajos y medios, sin alta frecuencia.
- En la salida de Band Pass la señal tan solo sale con los armónicos medios.

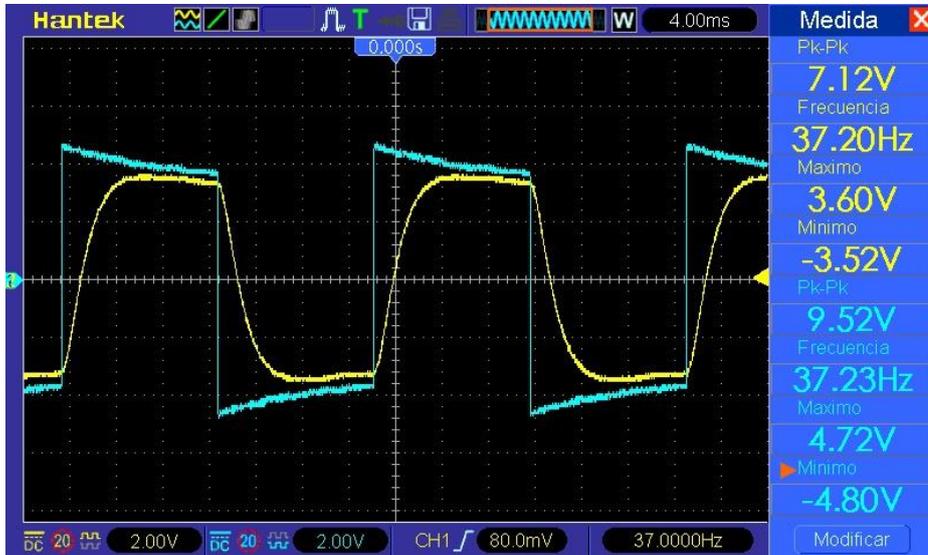


Figura 56 - VCF con frecuencia de corte a la mitad, salida LP

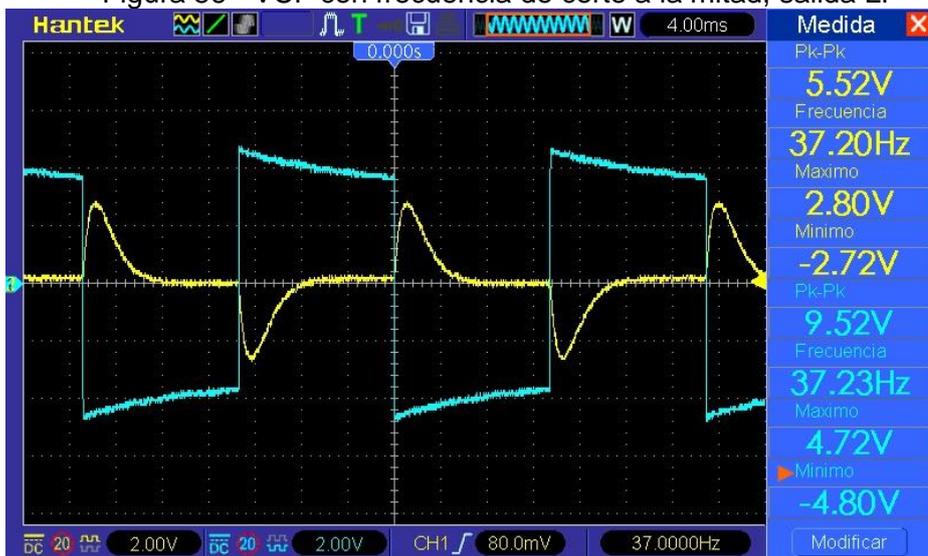


Figura 57 - VCF con frecuencia de corte a la mitad, salida BP

En la figura 58 y 59 podemos observar la misma señal que en la Figura 56 y 57, pero el ajuste de resonancia a la mitad, que hace que se le suma un componente a la señal causando sobreoscilación unos breves instantes.

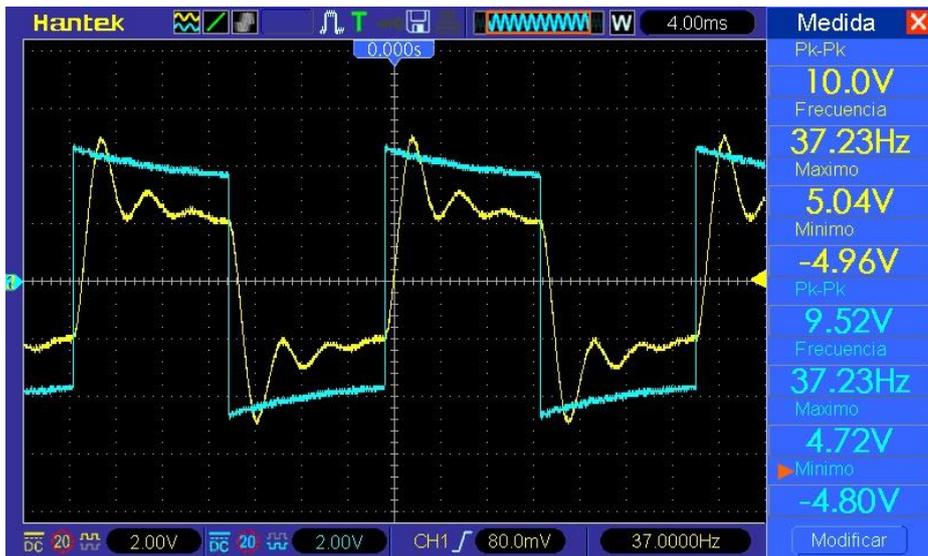


Figura 58 - VCF con frecuencia de corte a la mitad, resonancia media, salida LP

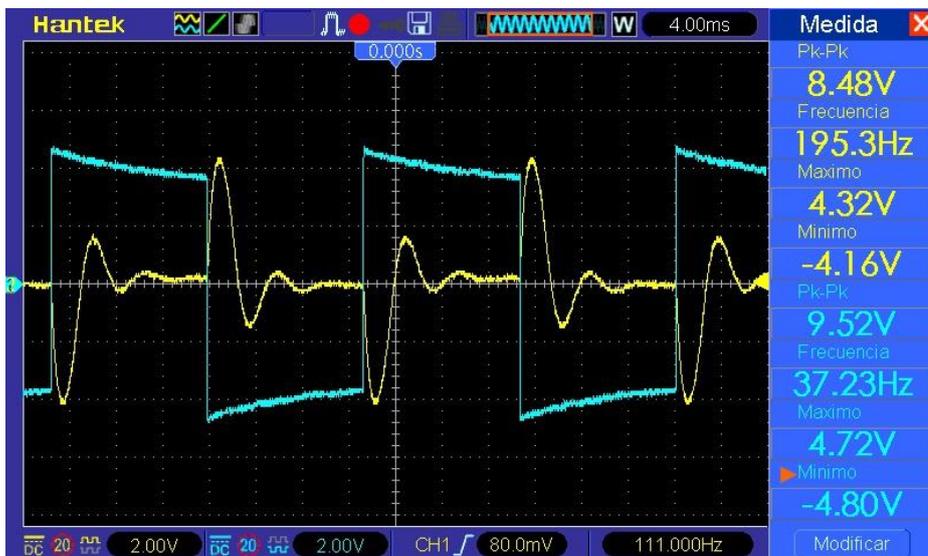


Figura 59 - VCF con frecuencia de corte a la mitad, resonancia media, salida BP

En la figura 60 podemos observar la misma señal que en la Figura 58, pero el ajuste de resonancia a la mitad, que hace que se le suma un componente a la señal causando sobreoscilación con intensidad.



Figura 60 - VCF con frecuencia de corte a la mitad, resonancia alta, salida LP

En la figura 61 y 62 podemos observar la señal filtrada estar el potenciómetro de ajuste de frecuencia de corte totalmente abierto y el ajuste de resonancia en posición 0.

- En la salida de Low Pass la señal sale con todos los contenidos armónicos de la entrada.
- En la salida de Band Pass la señal tan solo sale con los armónicos altos.

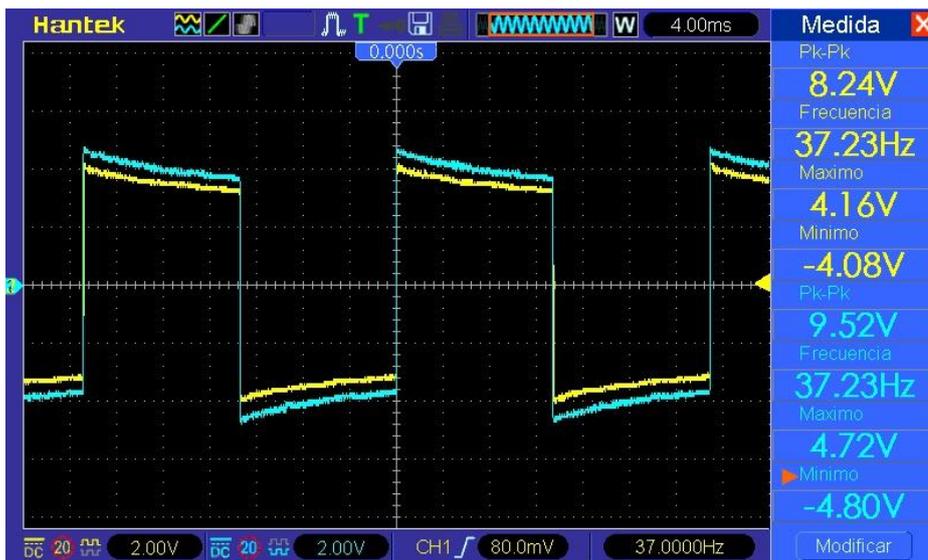


Figura 61 - VCF con frecuencia de corte máxima, salida LP

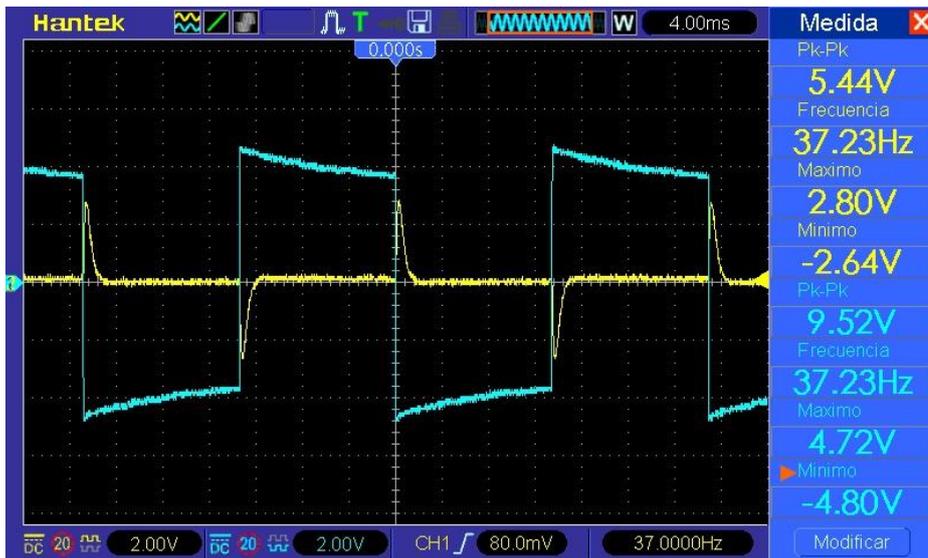


Figura 62 - VCF con frecuencia de corte máxima, salida BP

En la figura 63 podemos observar el diagrama de bode de la salida LP cuando el control de la frecuencia de corte está a la mitad y la resonancia a 0, que en la figura 64 la frecuencia de corte está a la mitad y la resonancia a 0.5.

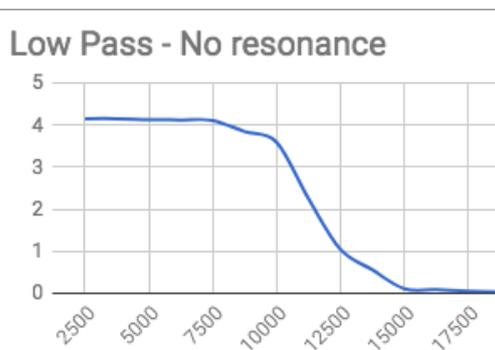


Figura 63 - Bode LP Resonancia 0



Figura 64 - Bode LP Resonancia 0.5

En la figura 65 podemos observar el diagrama de bode de la salida BP cuando el control de la frecuencia de corte está a la mitad y la resonancia a 0, que en la figura 66 la frecuencia de corte está a la mitad y la resonancia a 0.5.

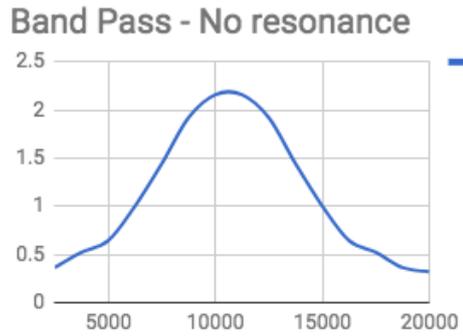


Figura 65 - Bode LP Resonancia 0

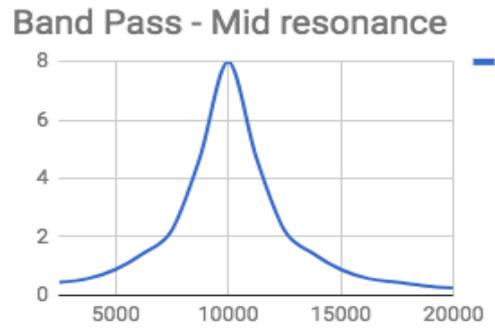


Figura 66 - Bode BP Resonancia 0.5

En la figura 63B podemos observar el diagrama de bode de la salida LP cuando el control de la frecuencia de corte está a un octavo del giro y la resonancia a 0, que en la figura 64B la frecuencia de corte está a la mitad y la resonancia a 0.5.

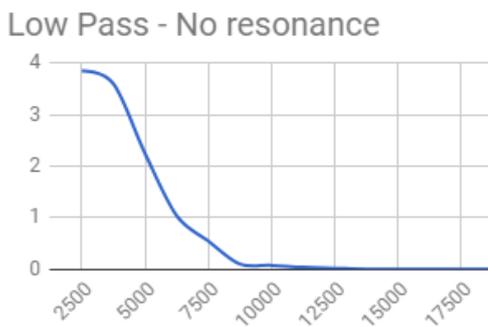


Figura 63B - Bode LP Resonancia 0

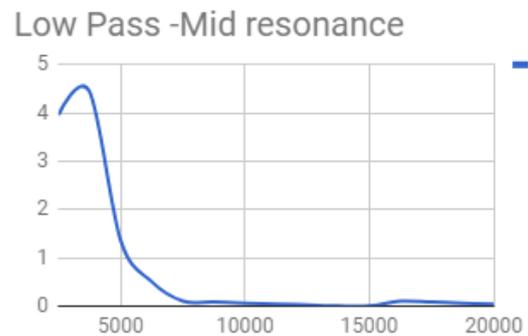


Figura 64B - Bode LP Resonancia 0.5

En la figura 65B podemos observar el diagrama de bode de la salida BP cuando el control de la frecuencia de corte está a un octavo del giro y la resonancia a 0, que en la figura 66B la frecuencia de corte está a la mitad y la resonancia a 0.5.

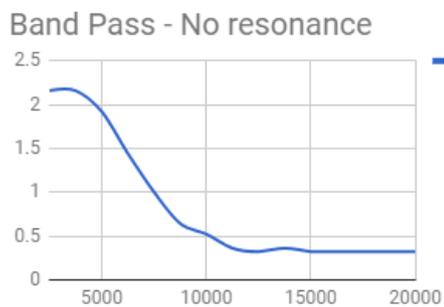


Figura 65B - Bode LP Resonancia 0

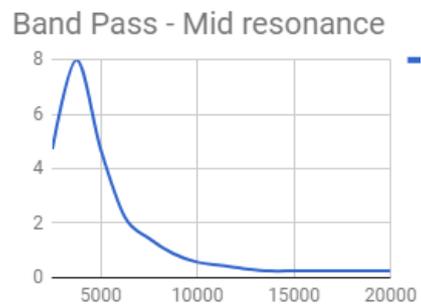


Figura 66B - Bode BP Resonancia 0.5

En la figura 63B podemos observar el diagrama de bode de la salida LP cuando el control de la frecuencia de corte está a siete octavos del giro y la resonancia a 0, que en la figura 64B la frecuencia de corte está a la mitad y la resonancia a 0.5.

Low Pass - No resonance

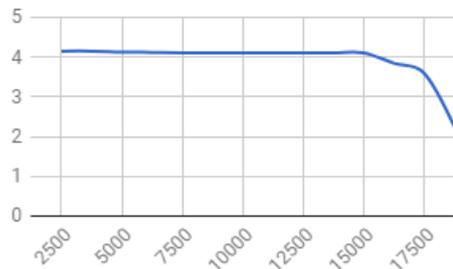


Figura 63B - Bode LP Resonancia 0

Low Pass -Mid resonance

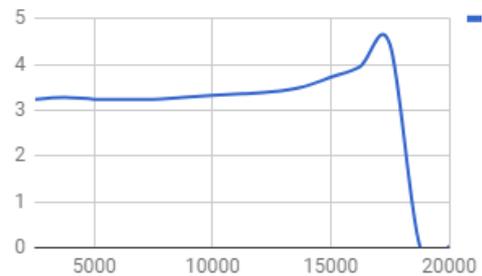


Figura 64B - Bode LP Resonancia 0.5

En la figura 65B podemos observar el diagrama de bode de la salida BP cuando el control de la frecuencia de corte está a siete octavos del giro y la resonancia a 0, que en la figura 66B la frecuencia de corte está a la mitad y la resonancia a 0.5.

Band Pass - No resonance

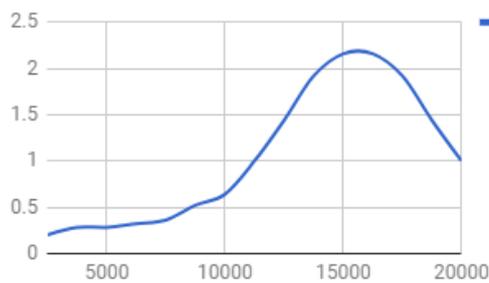


Figura 65B - Bode LP Resonancia 0

Band Pass - Mid resonance

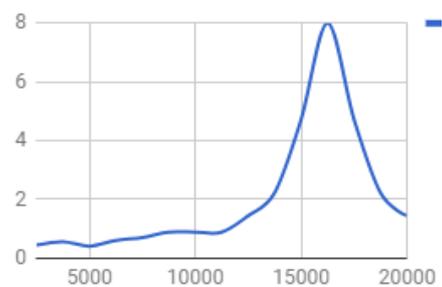


Figura 66B - Bode BP Resonancia 0.5

En la figura 67 podemos observar la señal de salida tanto en LP como BP, estando el potenciómetro de ajuste de frecuencia de corte a mitad y el ajuste de resonancia al máximo, sin tener una entrada de señal, esto es la auto oscilación del filtro, y la frecuencia de esta señal se ajusta con el potenciómetro de frecuencia de corte. Esta señal es senoidal debido a su naturaleza.

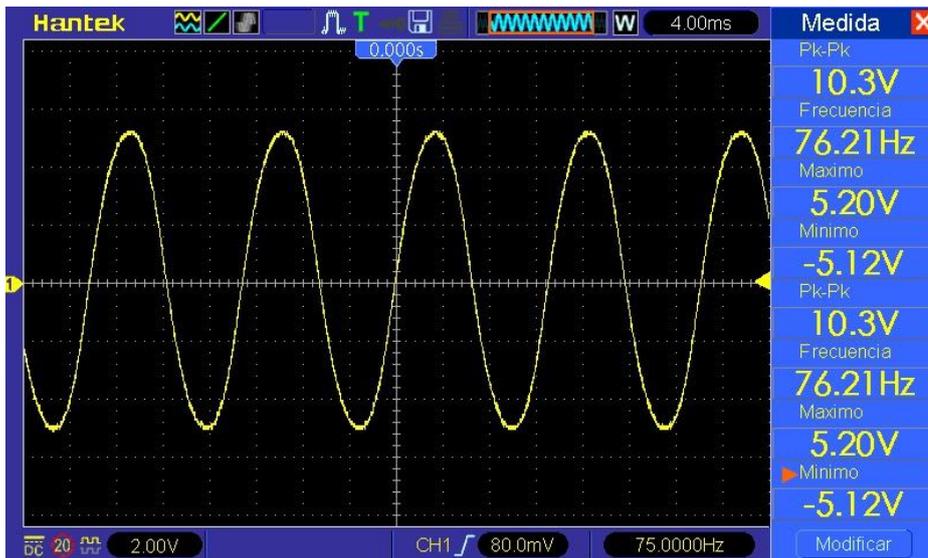


Figura 67 - VCF en auto oscilación

Módulos digitales: En este caso, el testeo se realiza de la misma forma que los módulos analógicos, pero no concluye con el testeo de hardware, igual que con los módulos analógicos, se procede a comprobar si el encendido es correcto, después se comprueba que ningún componente se caliente por encima de lo normal, y si estas pruebas son superadas, se comprobará si se pueden programar los microcontroladores. En caso de que se puedan programar los microcontroladores sin ningún problema, pasaremos al testeo de software.

10.3. Testeo de software

El testeo de software se realiza tan solo para los módulos digitales, ya que se analizan las funciones que residen en el código del dispositivo, en caso de ser un módulo analógico, ya se han analizado sus funciones en el testeo de hardware.

VCO: Genera una señal oscilante de una frecuencia comprendida entre [20-1000]Hz y de amplitud [-5,5]V, ajustable en frecuencia y forma de onda.

En la figura 68 podemos observar la señal de salida, en este caso es una señal diente de sierra a la frecuencia mínima de funcionamiento, también se observa que cumple las especificaciones de amplitud.

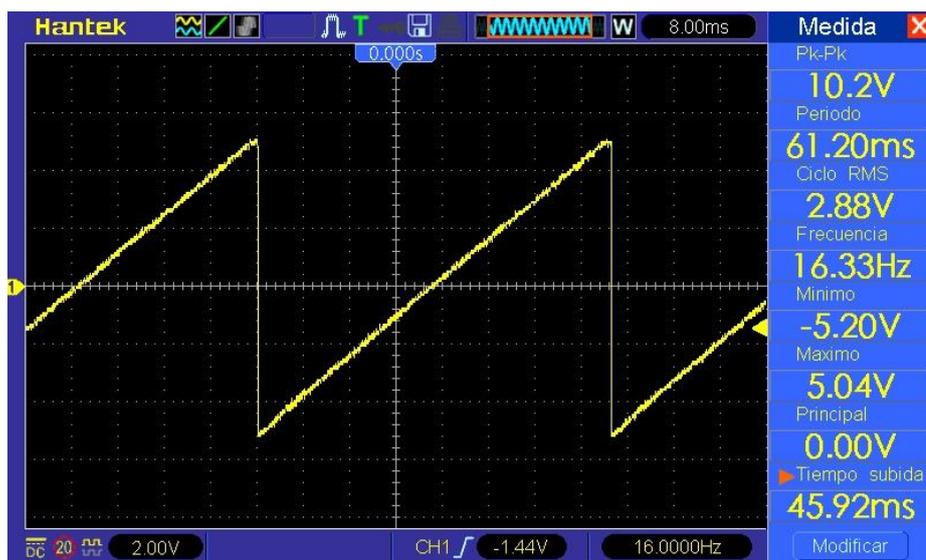


Figura 68 - Señal de salida del VCO a la mínima frecuencia

En la figura 69 podemos observar la señal de salida, en este caso es una señal cuadrada a 880Hz, esta onda corresponde con un LA de la 4ª octava.

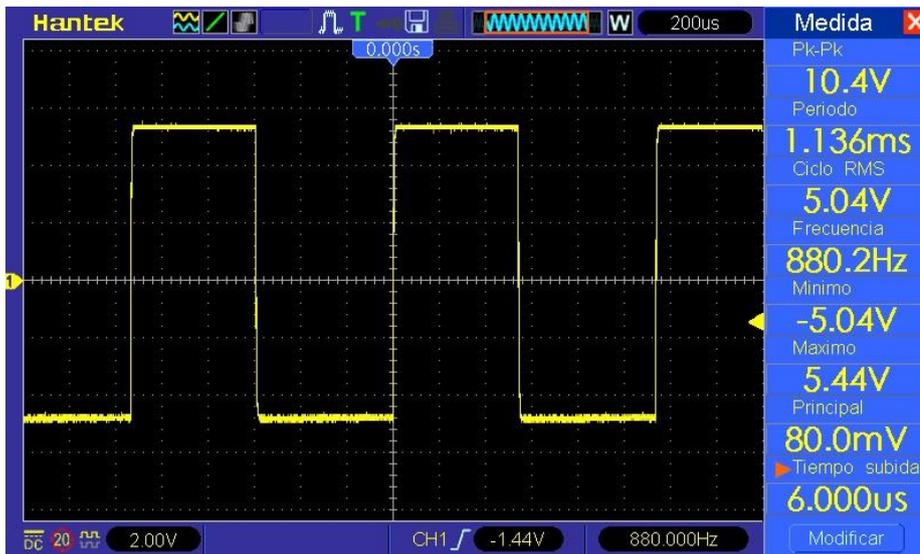


Figura 69 - Señal de salida del VCO a 880Hz

En la figura 70 podemos observar la señal de salida, en este caso es una señal compleja a más de 1kHz, comprobando el límite superior de los requisitos.

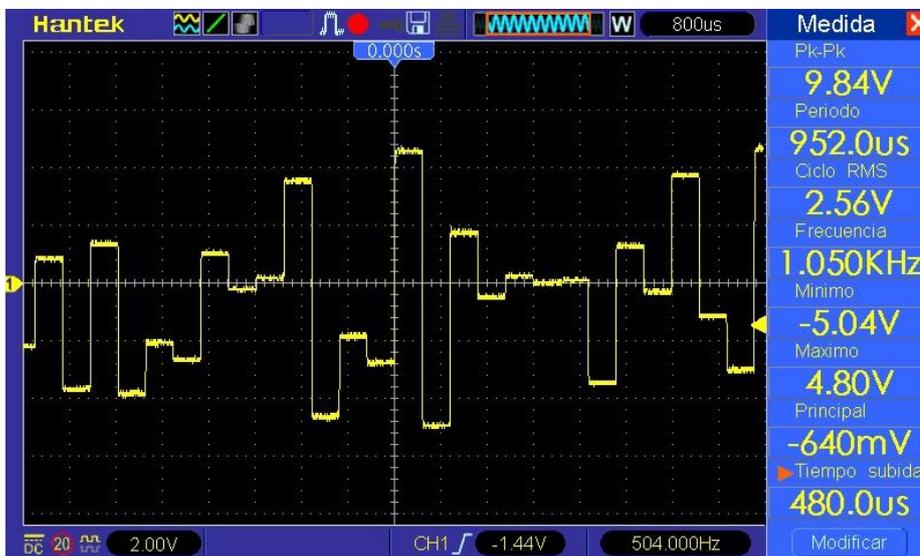


Figura 70 - Señal de salida del VCO a 1kHz

En la figura 71 podemos observar la señal de salida, en este caso es una señal cuadrada, modulada en frecuencia (FM) por una señal de diente de sierra.

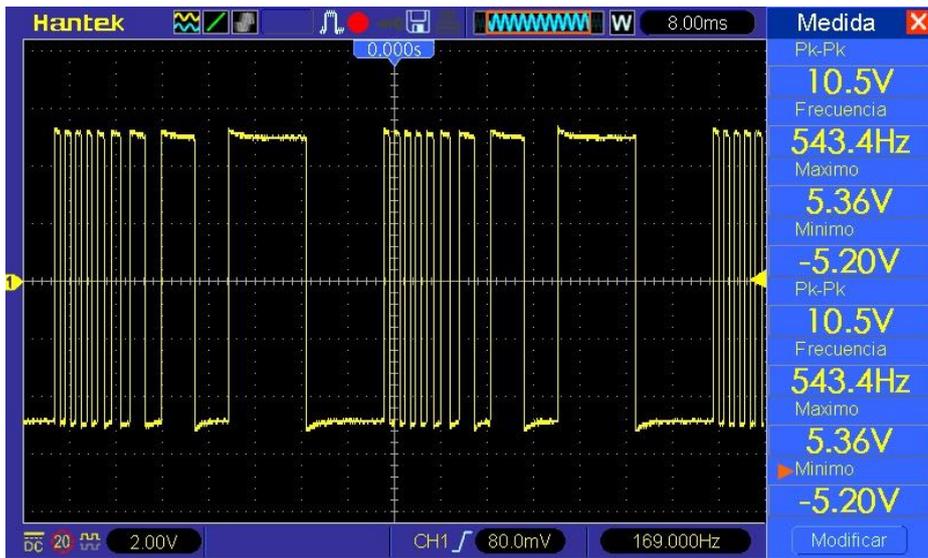


Figura 71 - Señal de salida del VCO con FM

LFO: Genera una señal oscilante de una frecuencia comprendida entre [0.01-100]Hz de amplitud en la salida unipolar [0,5]V y en la bipolar [-2.5,2.5]V ajustable en frecuencia y forma de onda. En este módulo se visualizan dos señales de forma simultánea, la salida **bipolar**, y la **unipolar**.

En la figura 72 podemos observar las señales de salida, en este caso es una señal senoidal, comprobando el límite inferior de frecuencia.

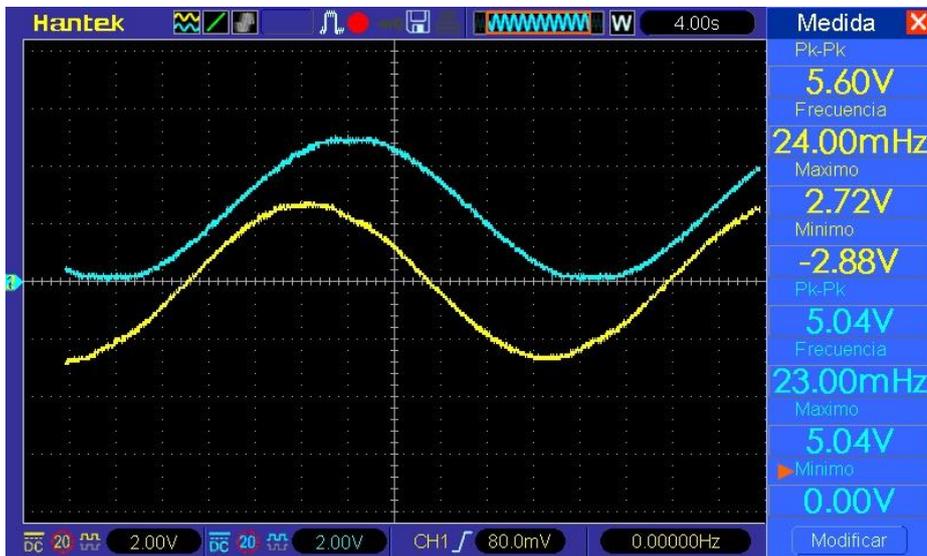


Figura 72 - Señal de salida del LFO de muy baja frecuencia

En la figura 73 podemos observar las señales de salida, en este caso es una señal triangular, comprobando el rango medio de frecuencia del LFO.

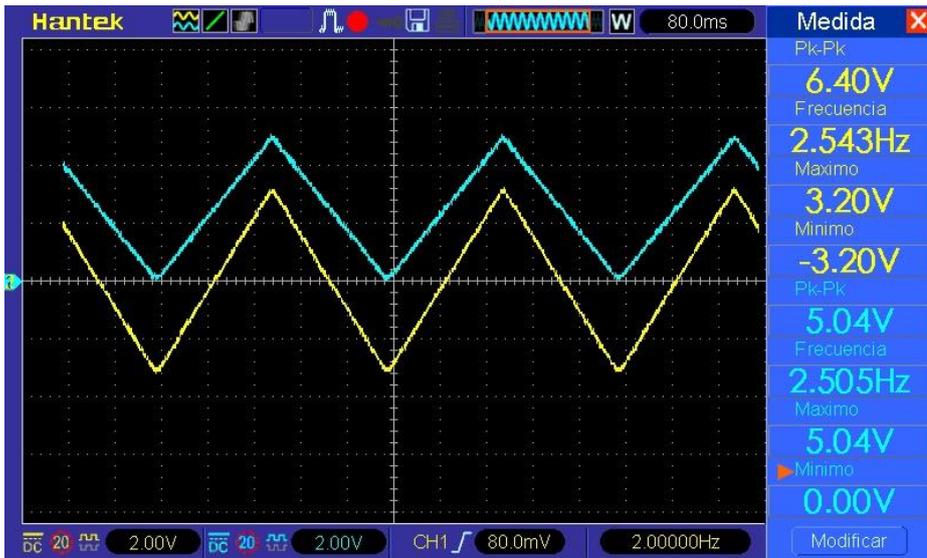


Figura 73 - Señal de salida del LFO de media frecuencia

En la figura 74 podemos observar las señales de salida, en este caso es una señal triangular, comprobando el rango alto de frecuencia del LFO.

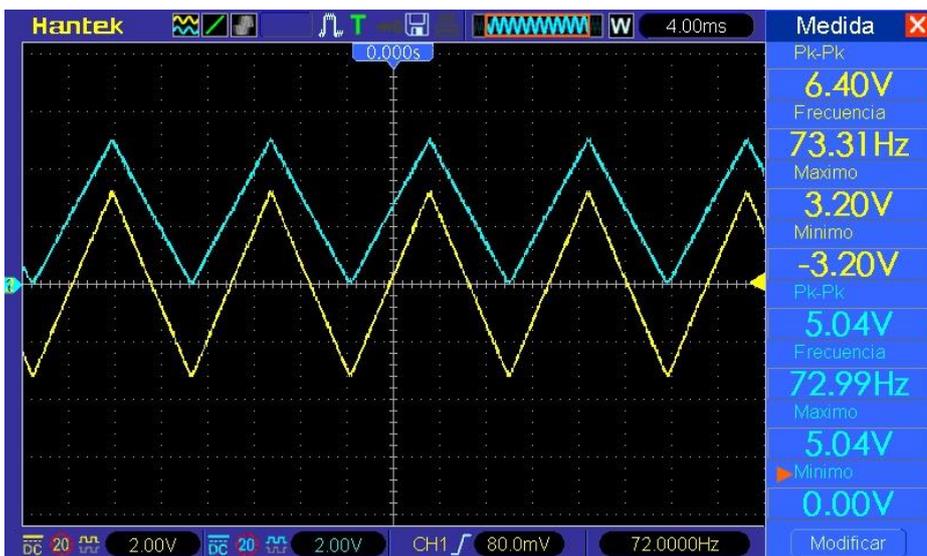


Figura 74 - Señal de salida del LFO de alta frecuencia

En la figura 75 podemos observar las señales de salida, en este caso es una señal diente de sierra, comprobando la modulación de frecuencia (FM) del LFO.

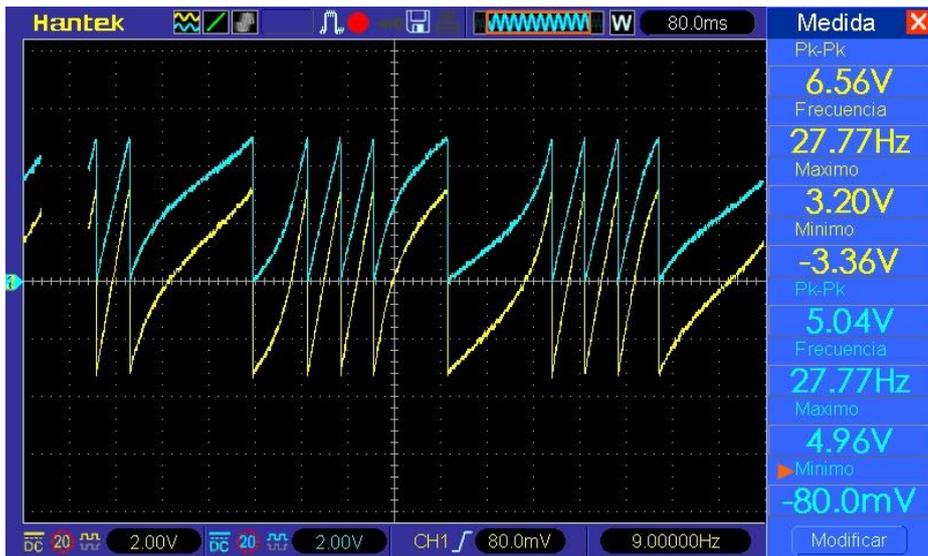


Figura 75 - Señal de salida del LFO modulada en frecuencia

En la figura 76 podemos observar las señales de salida, en este caso es una señal de ruido, comprobando la modulación de frecuencia (FM) del LFO.



Figura 76 - Señal de salida del LFO ruido modulado en frecuencia

ADSR: Generar una señal envolvente para dar expresividad a los sonidos, hacer que las señales de control de voltaje adquieran una forma más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo, con posibilidad de retrigger.

En la figura 77 podemos observar la salida del envolvente, en este caso es una señal de pulso extremadamente corta, el interruptor selector de velocidad se ha posicionado en la opción rápida y los ajustes con el mínimo tiempo posible. Esta imagen sirve para validar la velocidad del generador de envolvente.

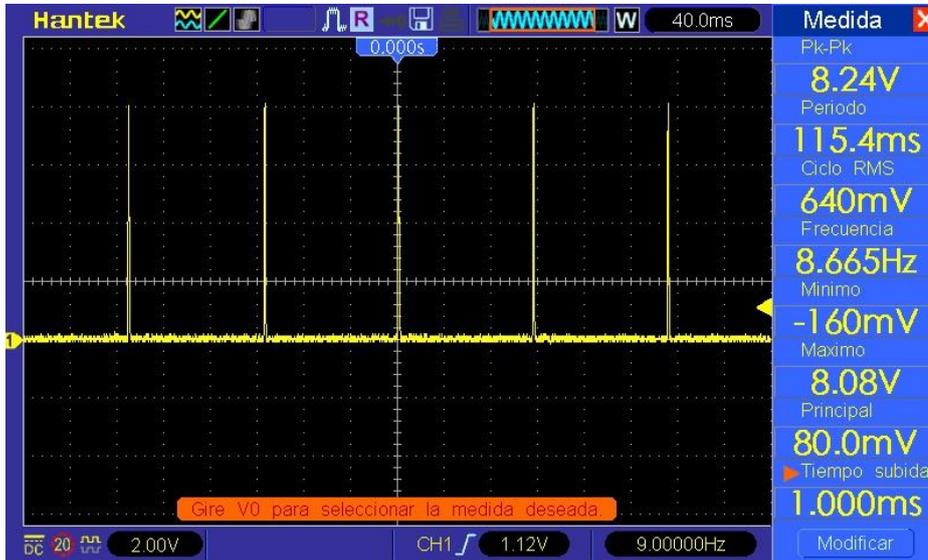


Figura 77 - Señal de salida del ADSR muy corta

En la figura 78 podemos observar la salida del ADSR, en este caso es una señal de pulso corta, el interruptor selector de velocidad se ha posicionado en la opción rápida y los ajustes con un tiempo medio, se puede ver como evoluciona con el tiempo.

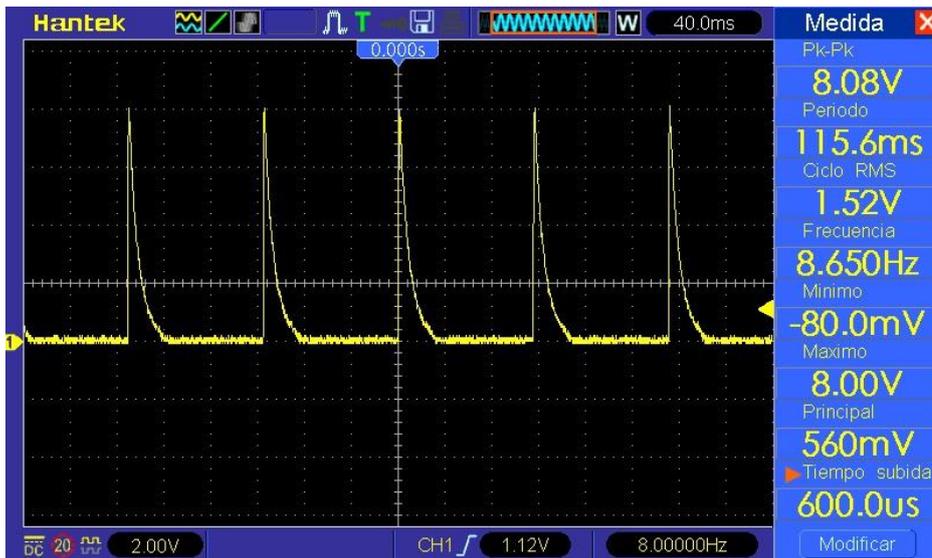


Figura 78 - Señal de salida del ADSR corta

En la figura 79 podemos observar la salida del ADSR, en este caso es una señal de pulso corta, el interruptor selector de velocidad se ha posicionado en la opción rápida y los ajustes con un tiempo largo, se puede ver como evoluciona con el tiempo.

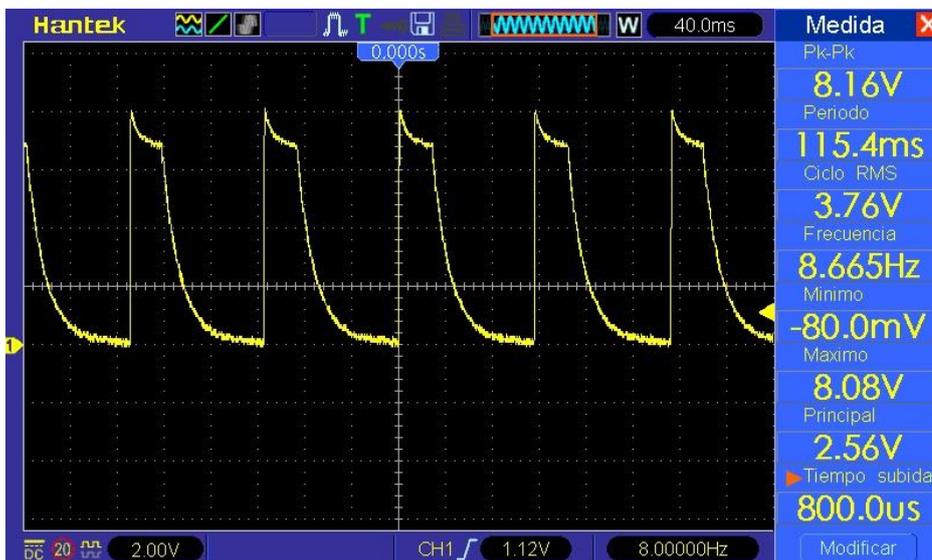


Figura 79 - Señal de salida del ADSR media

En la figura 80 podemos observar la salida del ADSR, en este caso es una señal de pulso largo, el interruptor selector de velocidad se ha posicionado en la opción lenta y los ajustes con un tiempo largo, se puede ver como evoluciona con el tiempo.

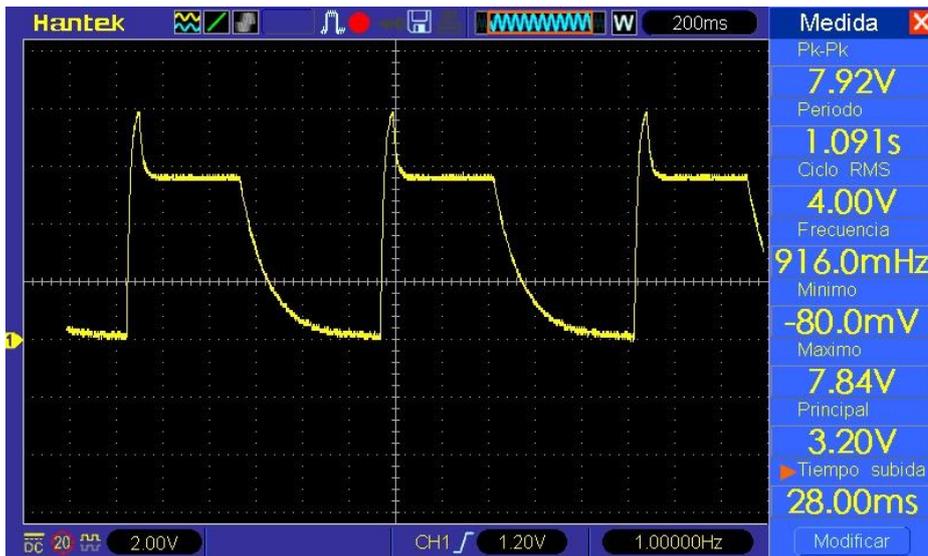


Figura 80 - Señal de salida del ADSR larga

En la figura 81 podemos observar la salida del ADSR, en este caso es una señal de pulso largo, el interruptor selector de velocidad se ha posicionado en la opción lenta y los ajustes con un tiempo largo, se puede ver como se activa el retrigger en medio de la envolvente.

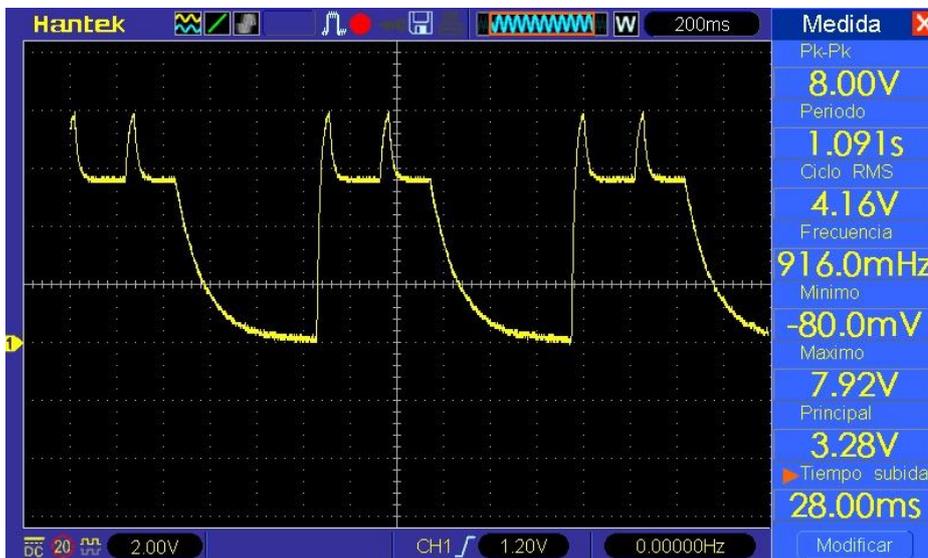


Figura 81 - Señal de salida del ADSR larga con retrigger

10.4. Testeo de elementos mecánicos

Los elementos mecánicos son muy importantes en estos módulos, ya que sirven para sujetar el dispositivo a los raíles de eurorack, y permiten el ajuste y movimiento de sus potenciómetros y botones. Este testeo se comprueba utilizando el módulo de forma natural en los raíles de Eurorack.



Figura 82 - Módulos montados en sus raíles

- **Comprobación de la sujeción en los raíles:** El aspecto más importante de este apartado es que estos módulos queden bien sujetos en los raíles, ya que van a estar expuestos al movimiento y su utilización. Esta prueba consiste en montar con tornillos M3 los módulos a los raíles y comprobar si existe algún tipo de holgura o balanceo. En caso negativo, se puede avanzar a la siguiente fase.
- **Comprobación de elementos mecánicos:** Ya comprobada su sujeción en el raíl, se pasa a comprobar la sujeción de los componentes en el panel:
 - Potenciómetros: Deben de estar sujetos con firmeza para garantizar un giro suave y agradable.
 - Conectores jack: Deben de estar sujetos con firmeza, ya que van a soportar multitud de conexiones y desgaste creado por el uso.
 - Botones: Deben de estar sujetos con firmeza, y accesibles al público, ya que van a tener que utilizarlos para cambiar las funciones de algunos módulos.

- LEDs: Deben de estar sujetos con firmeza, y visibles al público, ya que nos van a informar con su color o luminosidad diferentes estados de sus funciones

Si los dispositivos han logrado superar todas las comprobaciones incluidas en esta validación del dispositivo, se puede decir que se ha realizado un gran trabajo, y que ahora solo queda disfrutar este sistema de síntesis.

11. Resumen del trabajo desarrollado:

En este apartado se comenta el trabajo previo a este proyecto, el porqué de su creación, la idea de negocio y el trabajo futuro.

11.1. Trabajo previo

Este proyecto es el resultado de una gran cantidad de horas investigando, trabajando y prototipando. No es la primera vez que afronto algo de esta temática, ya que mi interés por los dispositivos sonoros ha existido siempre.

Ya hace 2 años, fabriqué mi primer dispositivo sonoro, el **Vaporsynth**, que es una versión de el sintetizador basado en arduino creado por Notes and Volts [32]



Figura 83 - Vaporsynth

Después del vaporsynth diseñé y fabriqué unas PCBs que sirven de shield para Arduino Uno, transformándolo en una caja de ritmos llamada **Minipops**, basada en el proyecto realizado por Jan Ostman en su blog [33]

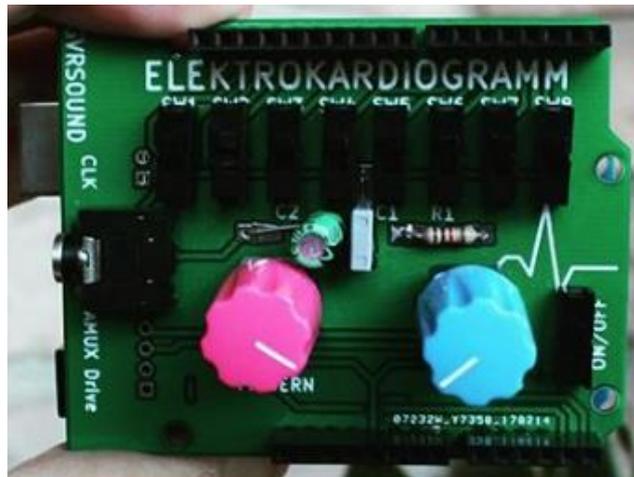


Figura 84 - Minipops drum machine

Gracias a mis compañeros de Makers UPV, tuve la oportunidad de organizar dos talleres donde se montó y programó un diseño completamente realizado por mí, **Ritmo**, es un secuenciador de ritmos y patrones electrónicos que se basa en el ATmega328.



Figura 85 - Ritmo Sequencer

Después de Ritmo, dediqué mi verano pasado a diseñar y programar Grano, un sintetizador granular basado en ATmega328, con teclado capacitivo y multitud de funcionalidades.

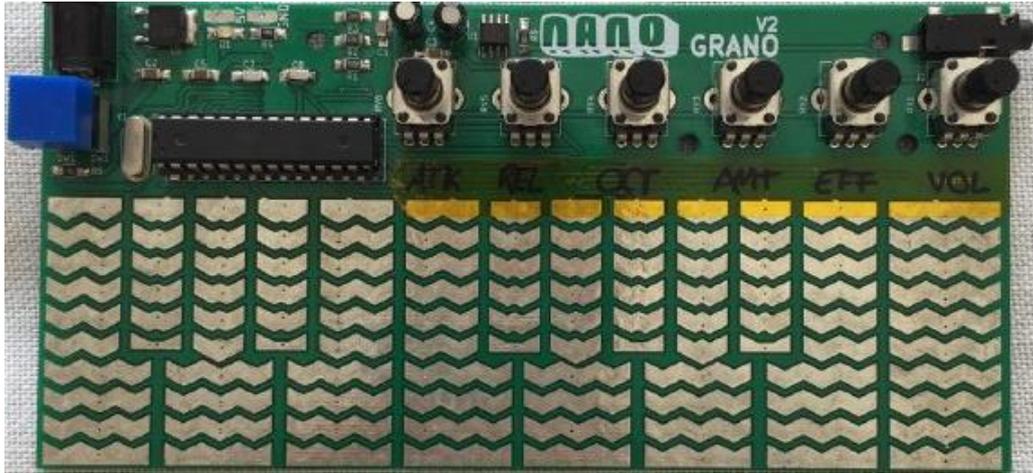


Figura 86 - Grano Synthesizer

Estos son los dispositivos sonoros que he fabricado antes de llegar a este proyecto, los cuales me han servido de inspiración y experiencia para afrontar el sintetizador modular con fuerza.

11.2. Sistema final

El propósito de este proyecto es el **Diseño e implementación de un sintetizador de audio modular basado en síntesis substractiva**. Una vez se ha comprobado que todo cumple las especificaciones, se ha procedido a montar el sistema completo, mostrado en la figura 87:



Figura 87 - Módulos montados en sus raíles

Este sistema incluye los 5 módulos del proyecto, dos de cada uno ya que se hicieron dos prototipos para asegurar el funcionamiento. Con el sistema montado, se puede asegurar su funcionamiento y la correcta comunicación entre módulos. Estos módulos no son los únicos en el sistema, ya que incluye varios módulos que tenía antes de empezar el proyecto, los cuales funcionan bien con los propio del proyecto. En la [Figura 88] se muestran los módulos del sistema en detalle.



Figura 88 - Módulos del sistema

11.3. Marca NANO

NANO es la marca que he creado para agrupar todos los proyectos sonoros que voy haciendo, como los dispositivos presentados en el apartado anterior y este mismo proyecto.



Figura 89 - Logo NANO

NANO se creó con la intención de en un futuro próximo poder comercializar los dispositivos diseñados por mí. Actualmente está en fase de crecimiento, tiene presencia en las redes sociales, la web está en construcción, y se publica constantemente en las redes para crear comunidad.

NANO es una iniciativa Open Source [34] licenciada bajo **CC BY-NC-SA 4.0** [35] esto quiere decir que los dispositivos se publican mostrando el código, esquemáticos y toda la documentación. Esta iniciativa ha gustado mucho a la gente por el hecho de aportar conocimiento.

11.4. Trabajo futuro

NANO no ha hecho más que comenzar, he utilizado este proyecto para dar un paso adelante y buscar una orientación comercial para los dispositivos creados. Los próximos pasos serán:

- **Buscar beta-testers para los módulos:** Personas encargadas de el testeo y prueba de estos módulos, para valorar su funcionamiento y si vale la pena salir a mercado con ellos.
- **Realizar los diseños definitivos:** Ya con el feedback de los beta-testers, y otras mejoras, realizar la producción ya preparada para ser comercializada
- **Darme de alta como autónomo:** para poder realizar ventas de forma legal.
- **Comercializar los módulos por todo el mundo.**

11.5. Conclusiones

11.5.1. Verificación de funcionamiento.

Lo más importante del proyecto, es que una vez montado se pueda comprobar que los módulos cumplen los requisitos marcados en el inicio del proyecto, por tanto se va a analizar cada módulo de forma independiente para aclarar su funcionamiento y cumplimiento de las especificaciones marcadas.

VCO: Genera una señal oscilante de una frecuencia comprendida entre [20-1000]Hz y de amplitud [-5,5]V, ajustable en frecuencia y forma de onda.

Este módulo ha sido el más complejo de poner en marcha, ya que han habido que realizar numerosos ajustes tanto de hardware como de software, sin embargo, se han conseguido con creces sus especificaciones fijadas, no sin antes sufrir frente al compilador para conseguir llegar al rendimiento adecuado. En el apartado **10.3. Testeo de software** se puede observar en las figuras como cumple estas especificaciones, ya que en la figura 68 cumple con la mínima frecuencia requerida, y en la figura 70 la máxima, mientras se puede ver su amplitud, también acorde a lo especificado.

LFO: Genera una señal oscilante de una frecuencia comprendida entre [0.01-100]Hz de amplitud en la salida unipolar [0,5]V y en la bipolar [-2.5,2.5]V ajustable en frecuencia y forma de onda.

Este módulo es muy similar al VCO tanto en hardware como en software, por tanto su puesta a punto ha sido muy semejante a la del VCO. Se ha conseguido alcanzar las especificaciones fijadas, no sin antes sufrir frente al compilador para conseguir llegar al rendimiento adecuado. En el apartado **10.3. Testeo de software** se puede observar en las figuras como cumple estas especificaciones, ya que en la figura 72 cumple con la mínima frecuencia requerida, y en la figura 74 la máxima, mientras se pueden ver sus dos salidas, la unipolar y bipolar, en la figura 75 también se puede ver la modulación de frecuencia, acorde a lo especificado.

ADSR: Genera una señal envolvente para dar expresividad a los sonidos, hacer que las señales de control de voltaje adquieran una forma más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo, con posibilidad de retrigger.

Este módulo es el único que incorpora el integrado ATmega328 para su funcionamiento. Se ha conseguido alcanzar las especificaciones fijadas, llegando sin problemas al rendimiento adecuado. En el apartado **10.3. Testeo de software** se puede observar en las figuras como cumple estas especificaciones, ya que en las figuras [77-81] se muestra su funcionamiento, acorde a lo especificado.

VCA: Amplifica o atenúa la señal con control de voltaje, ajustando la ganancia requerida. Este módulo es completamente analógico, por tanto su funcionamiento cae sobre el apartado del testeo de hardware. En el apartado **10.2. Testeo del hardware** se puede observar en las figuras como cumple estas especificaciones, ya que en las figuras [49-53] se muestra su funcionamiento, acorde a lo especificado. Este módulo ha sido el menos costoso desde el punto de vista del diseño.

VCF: Filtra la señal de audio en torno a frecuencias comprendidas entre [20-20000]Hz, pudiendo ajustar la resonancia y frecuencia de corte del filtro y tener dos salidas simultáneas, Low Pass y Band Pass. Este módulo es completamente analógico, por tanto su funcionamiento cae sobre el apartado del testeo de hardware. En el apartado **10.2. Testeo del hardware** se puede observar en las figuras como cumple estas especificaciones, ya que en las figuras [54-67] se muestra su funcionamiento, acorde a lo especificado. El diseño de este módulo ha sido complejo para ser un módulo analógico debido a la gran cantidad de componentes que incluye.

11.5.2. Conclusión personal.

Este proyecto ha recorrido un largo camino, empezó a finales de Octubre cuando fui a hablar con Salvador para presentarle mi idea, nos encargamos de darle forma y lo convertimos en un TFG muy prometedor. Obtuve el sí por respuesta a hacerlo, y ahí empecé con todo el prototipado, diseño, y fabricación.

El resultado final del proyecto es excelente, cumple las expectativas con creces, y lo mejor es que sigue mejorando día a día, no es algo estático. Lo que más ha costado de este proyecto es el dinero para llevarlo a cabo, en total han sido 850€ en materiales, lo mejor del proyecto son los dispositivos creados, que en un futuro se comercializarán como parte de la marca NANO.

Esto no es un final, es un comienzo.

Este proyecto ha sido el primer paso hacia algo más serio, hacia la creación de un posible negocio real al cual voy a volcar mi tiempo en los próximos meses, y si va bien, años, por tanto le he puesto todo de mi parte a este TFG, sabiendo que no solo era un TFG, era algo más, es un negocio real.

Para finalizar, quiero aclarar que voy a continuar con este proyecto mucho más allá del TFG, y que me voy a seguir esforzando para superar retos como este y los que puedan venir.

12. Bibliografía

Libros:

PUSH TURN MOVE - Kim Bjorn / Denmark / 2017

MAKE: ANALOG SYNTHESIZERS - Ray Wilson / Maker Media / EEUU 2013

THE SYNTHESIZER - Mark Vail / Oxford University Press / United Kingdom 2014

ANALOG SYNTHESIZERS - Mark Jenkins / Routledge / EEUU 2009

Enlaces:

[1] Sintetizadores analógicos y digitales:

Fecha: 19/11/2017

<http://karateaudio.com/sintetizador-analogo-vs-digital/>

[2] Síntesis aditiva:

Fecha: 19/11/2017

https://es.wikipedia.org/wiki/S%C3%ADntesis_aditiva

[3] Síntesis sustractiva:

Fecha: 20/11/2017

https://es.wikipedia.org/wiki/S%C3%ADntesis_substractiva

[4] Síntesis por amplitud modulada:

Fecha: 21/11/2017

https://es.wikipedia.org/wiki/Amplitud_modulada

[5] Síntesis por frecuencia modulada:

Fecha: 21/11/2017

https://es.wikipedia.org/wiki/Frecuencia_modulada

[6] Síntesis por modelado físico:

Fecha: 23/11/2017

https://es.wikipedia.org/wiki/S%C3%ADntesis_por_modelado_f%C3%ADsico

[7] Síntesis granular:

Fecha: 23/11/2017

https://es.wikipedia.org/wiki/S%C3%ADntesis_granular

[8] Introducción a la síntesis sustractiva:

Fecha: 27/11/2017

<https://www.hispasonic.com/tutoriales/introduccion-sintesis-sustractiva/73>

[9] Mercado de los sintetizadores:

Fecha: 31/11/2017

<https://www.technavio.com/report/global-consumer-electronics-music-synthesizers-market>

[10] Foro de eurorack:

Fecha: 2/12/2017

<https://www.muffwiggler.com/forum/viewtopic.php?t=803&highlight=>

[11] Módulos de eurorack y sus cifras:

Fecha: 6/12/2017

<https://www.modulargrid.net/e/modules>

[12] Grupo de Facebook de aficionados al sonido y DIY:

Fecha: 10/12/2017

<https://www.facebook.com/groups/cacharraje/>

[13] Foro destinado a compartir conocimiento acerca de proyectos sonoros DIY:

Fecha: 13/12/2017

<https://www.muffwiggler.com/forum/viewforum.php?f=17>

[14] Manual de características mecánicas de Eurorack:

Fecha: 16/12/2017

http://www.doepfer.de/a100_man/a100m_e.htm

[15] Manual de características eléctricas de Eurorack:

Fecha: 18/12/2017

http://www.doepfer.de/a100_man/a100t_e.htm

[16] Manual de la fuente de alimentación uZeus de Eurorack:

Fecha: 20/12/2017

<http://www.tiptopaudio.com/manuals/uzeuswfbb.pdf>

[17] Web del fabricante Mutable Instruments:

Fecha: 3/1/2018

<https://mutable-instruments.net/>

[18] Web del fabricante Make Noise:

Fecha: 3/1/2018

<http://www.makenoisemusic.com/>

[19] Web del fabricante Xaoc Devices:

Fecha: 3/1/2018

<http://xaocdevices.com/main/>

- [20] Cifras del mercado de los sintetizadores analógicos:
Fecha: 5/1/2018
<https://www.hispasonic.com/reportajes/vivimos-burbuja-sintetizadores-analogicos/43040>
- [21] Cifras del mercado de los sintetizadores clásicos:
Fecha: 15/1/2018
<http://www.sintemania.com/sintes-clasicos-y-la-realidad-de-sus-cifras/>
- [22] Manual de los railes zRails de Eurorack:
Fecha: 17/1/2018
<http://www.tiptopaudio.com/manuals/z-rails.pdf>
- [23] Web del programa de diseño de circuitos KiCad:
Fecha: 23/1/2018
<https://es.wikipedia.org/wiki/KiCad>
- [24] Web del entorno de desarrollo Arduino:
Fecha: 26/1/2018
<https://es.wikipedia.org/wiki/Arduino>
- [25] Web del entorno de desarrollo Keil:
Fecha: 30/1/2018
<http://www2.keil.com/mdk5/uvision/>
- [26] Tutorial de interconexión de señales entre sintetizadores:
Fecha: 3/2/2018 <https://www.hispasonic.com/tutoriales/sintesis-17-interconexion-control-analogicos/39017>
- [27] Información sobre la tecnología SMD
Fecha: 5/2/2018
https://en.wikipedia.org/wiki/Surface-mount_technology
- [28] Creación de una lista de redes en KiCad:
Fecha: 10/2/2018
<http://kicad.txplore.com/index-p=115.html>
- [29] Comprobación de las reglas de diseño en KiCad:
Fecha: 13/2/2018
<http://kicadhowto.wikidot.com/co1drc>
- [30] Información sobre los archivos Gerber:
Fecha: 18/2/2018
https://en.wikipedia.org/wiki/Gerber_format
- [31] Información sobre los microcontroladores de la gama STM32F334:
Fecha: 25/2/2018 <http://www.st.com/en/evaluation-tools/stm32-mcu-nucleo.html?querycriteria=productId=LN1847>

[32] Proyecto del sintetizador granular de Notes and Volts

Fecha: 3/4/2018

<https://www.youtube.com/watch?v=NTob27IOpcU>

[33] Proyecto de la caja de ritmos Minipops de Jan Ostman:

Fecha: 6/4/2018

<https://janostman.wordpress.com/the-o2-source-code/>

[34] Información sobre Open Source:

Fecha: 8/4/2018

<https://opensource.org/osd>

[35] Información sobre la licencia BY NC SA:

Fecha: 12/4/2018

<https://creativecommons.org/licenses/by-nc-sa/4.0/>



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño e implementación de un sintetizador de audio modular basado en síntesis substractiva

PRESUPUESTOS - TFG

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Gutiérrez-Ravé Olmos, Jorge

Tutor: Coll Arnau, Salvador

Índice

1. Introducción	5
2. Presupuesto prototipo	7
2.1. Costes materiales del prototipo	7
2.2. Costes de mano de obra	19
2.3. Presupuesto de ejecución material	19
3. Presupuesto 50 unidades	21
3.1. Costes materiales del prototipo	22
3.2. Costes de mano de obra	32
3.3. Presupuesto de ejecución material	33
3.4. Beneficio industrial	33
3.5. Coste total 50 unidades	33
3.6. Precio de venta unitario	33
3.7. Posición en el mercado	34

1. Introducción

En este documento se estimarán los costes de ejecución de los módulos del proyecto (VCO, LFO, ADSR, VCA y VCF) considerando tanto los costes materiales del circuito como los costes de personal para el montaje y embalaje del dispositivo.

De entre las tres formas típicas de presupuestar se optará por el presupuesto por **costes de naturaleza** debido a su sencillez de cálculo y que es canon en los proyectos en los que se desarrollan dispositivos electrónicos.

Se realizará el cálculo de los costes materiales en el que se incluirá una tabla con los componentes utilizados en el circuito, seguido de una tabla con los costes de mano de obra.

Tras realizar el presupuesto del prototipo, se seguirá un proceso similar para obtener el presupuesto de una partida de 50 unidades con el fin de estudiar la viabilidad económica del producto. Para finalizar se aplicarán porcentajes de beneficio industrial, medios auxiliares y el IVA para obtener el coste total del lote así como el precio unitario final.

2. Presupuesto prototipo

En este punto se obtendrá el presupuesto correspondiente al desarrollo de los prototipos de los módulos del proyecto (VCO, LFO, ADSR, VCA y VCF) del sistema de síntesis sustractiva de forma individual. Se tendrán en cuenta los costes materiales así como los costes de mano de obra asociados al desarrollo del proyecto en su totalidad.

2.1. Costes materiales del prototipo

Este apartado concreta los costes materiales de cada módulo, dejando constancia de sus impuestos y los costes derivados de mermas y roturas.

VCO

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
C4,C2,C1	3	1uF	VJ1206Y105MXQTW1BC	Unpolarized capacitor	0.246
C5	1	1nF	VJ1206Y102MXQTW1BC	Unpolarized capacitor	0.082
C6,C7	2	47u	VE-470M1CTR-0505	Polarised capacitor	0.324
C8,C10	2	22pF	VJ1206A220KXQPW1BC	Unpolarized capacitor	0.164
C9	1	4.7uF	M865080640004	Polarised capacitor	0.16
C13,C3	2	1uF	M865060640001	Polarised capacitor	0.3
C14	1	10nF	VJ1206Y103MXQTW1BC	Unpolarized capacitor	0.082
C15,C11,C12	3	100nF	C1206C104K5RACTU	Unpolarized capacitor	0.27
D1,D2	2	D	1N4148W-7-F	Diode	0.278
D3	1	5.1V	BZT52C5V1-E3-08	Zener Diode	0.23
D4	1	3.3V	BZT52C3V3-E3-18	Zener Diode	0.23
D5	1	LED_RG B_5MM	WP154A4SUREQBFZGW	LED RGB, Common Cathode	1.46
J1	2	02x05	M20-9990546	Generic connector 02x05	0.328
J2,J3,J4,J5	4	Jack-2	Thonkiconn	2-pin audio jack	2
J6	1	Conn_01 x06	M20-9770646	Generic connector 01x06	0.205
Q1	1	MMBT39 04	MMBT3904	NPN Transistor	0.108
R2	1	100k	RC1206FR-07100KL	Resistor	0.082
R4	1	22k	RC1206FR-0722KL	Resistor	0.082
R5	1	15k	RC1206FR-0715KL	Resistor	0.082
R6	1	33k	RC1206FR-0733KL	Resistor	0.082
R8	1	6k8	RC1206FR-076K8L	Resistor	0.082
R10,R1,R17,R9, R3,R11,R12,R13, R7	9	10k	RC1206FR-0710KL	Resistor	0.738
R14,R15,R16	3	100	RC1206FR-07100L	Resistor	0.246
RV21,RV2,RV3	3	10K	Alpha D Shaft 10K	Potentiometer	5.1
SW1	1	SW_Pus	TL1105KF160Q	Push button switch	0.238

Presupuesto

		h			
U1,U3	2	TL072	TL072CDR	Dual Operational Amplifiers	0.968
U2	1	L7805	L7805CDT-TR	Voltage Regulator 5V,	0.607
U4	1	TC1262-33	TC1262-3.3VDBTR	Voltage Regulator 3.3V,	0.508
U5	1	STM32F334K8	STM32F334K8T6	Microcontroller	3.56
Y1	1	8MHz	LFXTAL003156Bulk	Two pin crystal	0.148
PCB	1	2 Layer	100x20mm	AIIPCB	3
PANEL	1	PCB	128x20mm	AIIPCB	3.00
KNOBS	3	1900h	D-Shaft	Thonk	1.86
				TOTAL	26.85

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Mermas y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Mermas y Roturas} = 26,85 * 0,02 = 0,54\text{€}$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Mermas y Roturas}$$

$$\text{Coste Total Materiales VCO} = 26.85 + 0.54 = 27.39\text{€}$$

Este precio no incluye IVA, el precio con IVA es el siguiente

$$\text{Coste Total Materiales VCO} + \text{IVA} = 27.39 * 1.21 = 33.14\text{€}$$

LFO

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
C1, C14	2	1uF	M865060640001	Polarised capacitor	0.3
C3, C4, C6	3	1uF	VJ1206Y105MXQW1BC	Unpolarized capacitor	0.246
C7, C8	2	47u	VE-470M1CTR-0505	Polarised capacitor	0.324
C9, C11	2	22pF	VJ1206A220KXQPW1BC	Unpolarized capacitor	0.164
C10	1	4.7uF	M865080640004	Polarised capacitor	0.16
C12, C13, C16	3	100nF	C1206C104K5RACTU	Unpolarized capacitor	0.27
C15	1	10nF	VJ1206Y103MXQW1BC	Unpolarized capacitor	0.082
D1, D2	2	D	1N4148W-7-F	Diode	0.278
D3	1	3.3V	BZT52C3V3-E3-18	Zener Diode	0.23
D4	1	LED_RG B_5MM	WP154A4SUREQBFZGW	LED RGB, Common Cathode	1.46
J1	1	02x05	M20-9990546	Generic connector 02x05	0.164
J2, J3, J4, J6	4	Jack-2	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	2
J5	1	Conn_01 x06	M20-9770646	Generic connector 01x06	0.205
Q1	1	MMBT39 04	MMBT3904	NPN Transistor	0.108
R1, R2, R4, R6, R9, R10, R12, R13, R17, R18	10	10k	RC1206FR-0710KL	Resistor	0.82
R3	1	15k	RC1206FR-0715KL	Resistor	0.082
R5, R8	2	5k6	RC1206FR-075K6L	Resistor	0.164
R7	1	33k	RC1206FR-0733KL	Resistor	0.082

Presupuesto

R11	1	3k	RC1206FR-073KL	Resistor	0.082
R14, R15, R16	3	100	RC1206FR-07100L	Resistor	0.246
R19	1	10M	RC1206FR-0710ML	Resistor	0.082
RV1, RV2	2	10K	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	VERTICAL: B10K – 10K Linear - D shaft	3.4
SW1, SW2	2	SW_Push	TL1105KF160Q	Push button switch	0.476
U1	1	L7805	L7805CDT-TR	Voltage Regulator 5V,	0.607
U2	1	TL074	TL074CDR	Quad Operational Amplifiers	0.484
U3	1	TC1262- 33	TC1262-3.3VDBTR	Voltage Regulator 3.3V,	0.508
U5	1	STM32F3 34K8	STM32F334K8T6	Microcontroller	3.56
Y1	1	8MHz	LFXTAL003156Bulk	Two pin crystal	0.148
PCB	1	2 Layer	100x30mm	AllPCB	3
PANEL	1	PCB	128x30mm	AllPCB	3.00
KNOBS	3	1900h	https://www.thonk.co.uk/shop/1900h-d/	Black - Glow Pointer (D-Shaft)	1.86
				TOTAL	24.59

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Mermas y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Mermas y Roturas} = 24,59 * 0,02 = 0,49\text{€}$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Mermas y Roturas}$$

$$\text{Coste Total Materiales LFO} = 24,59 + 0,49 = 25,08\text{€}$$

Este precio no incluye IVA, el precio con IVA es el siguiente

$$\text{Coste Total Materiales LFO} + \text{IVA} = 25,08 * 1,21 = 30,35\text{€}$$

ADSR

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
PCB	1	2 Layer	100x20mm	AllPCB	1.5
PANEL	1	Acryl	128x30mm	AllPCB	3.00
KNOBS	4	1900h	https://www.thonk.co.uk/shop/1900h-d/	Black - Glow Pointer (D-Shaft)	2.48
RV1 RV2 RV3 RV4	4	10k	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	Alpha Potentiometer	6.4
J3 J4 J5	3	Audio-Jack-2	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	1.35
C1 C2	2	47u	VE-470M1CTR-0505	Polarised Capacitor	0.324
C3 C4 C6	3	0.1uF	C1206C104K5RACTU	Unpolarised Capacitor	0.255
C7 C8	2	22p	VJ1206A220KXQPW1B C	Unpolarised Capacitor	0.18
D1 D2	2	D	1N4148W-7-F	Diode	0.29
D3	1	LED	LEDs:LED_D3.0mm	LED Diode	0.085
J1	1	Conn_02 x05_Odd _Even	M20-9990546	Generic connector 02x05	0.34
J2	1	Conn_01 x08	M20-9990534	Generic connector 01x08	0.255
Q1 Q2	2	Q_NPN_ BEC	MMBT3904	Transistor NPN	0.306
R1	1	100	RC1206FR-07100L	Resistor	0.085
R10 R11 R12 R13 R14 R2 R3 R4 R5 R6 R9	11	10k	RC1206FR-07100KL	Resistor	0.22
R7	1	6k	RC1206FR-076KL	Resistor	0.085
R8	1	0	RC1206FR-070L	Resistor	0.085
U1	1	MCP472 5	MCP4725	DAC Converter	0.901
U2	1	L7805	L7805CDT-TR	Linear Regulator	0.499

Presupuesto

U3	1	TL072	TL072CDR	Dual Operational Amplifiers	0.502
U4	1	ATMEGA 328P-AU	ATMEGA328P-AU	Microcontroller	1.76
Y1	1	Crystal	LFXTAL003156Bulk	Crystal	0.135
				TOTAL	21.04

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Mermas y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Mermas y Roturas} = 21,04 * 0,02 = 0,42\text{€}$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Mermas y Roturas}$$

$$\text{Coste Total Materiales ADSR} = 21,04 + 0,42 = 21.46\text{€}$$

Este precio no incluye IVA, el precio con IVA es el siguiente

$$\text{Coste Total Materiales ADSR} + \text{IVA} = 21.46 * 1.21 = 25.97\text{€}$$

VCA

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
PCB	1	2 Layer	100x30mm	AIIPCB	3
PANEL	1	Acryl	128x30mm	AIIPCB	3.00
KNOBS	2	1900h	https://www.thonk.co.uk/shop/1900h-d/	Black - Glow Pointer (D-Shaft)	1.24
RV1 RV3	2	100k	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	Alpha Potentiometer	3.4
RV2 RV4	2	100k	https://www.thonk.co.uk/shop/ttpots/	Song Huei Potentiometer	2
J1 J2 J4 J5 J6 J7 J8 J9	8	Jack-2	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	4
C1 C2 C3 C4	4	0.1uF	C1206C104K5RACTU	Unpolarized capacitor	0.36
D1 D2	2	D	1N4148W-7-F	Diode	0.278
D3 D4	2	LED	BZT52C5V1-E3-08	LED Diode	0.46
J3	2	02x05	M20-9990546	Generic connector 02x05	0.328
R1 R12 R15 R18 R19 R21 R4 R5 R6 R8	10	100k	RC1206FR-07100KL	Resistor	0.82
R10 R23	2	220k	RC1206FR-07220KL	Resistor	0.164
R11 R24	2	30k	RC1206FR-0730KL	Resistor	0.164
R16 R25	2	10k	RC1206FR-0710KL	Resistor	0.164
R17 R26	2	1k	RC1206FR-071KL	Resistor	0.164
R13 R14 R2 R20 R3 R7	6	22k	RC1206FR-0722KL	Resistor	0.492
R22 R9	2	220	RC1206FR-07220L	Resistor	0.164
U1 U3	2	TL072	TL072CDR	Dual Operational Amplifiers	0.968

Presupuesto

U2		LM1370 1 0	LM13700MX/NOPB	Dual Transconductance Amplifiers	0.951
				TOTAL	22.117

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Merms y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Merms y Roturas} = 22,117 * 0,02 = 0,44\text{€}$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Merms y Roturas}$$

$$\text{Coste Total Materiales VCA} = 22,117 + 0,44 = 22.557\text{€}$$

Este precio no incluye IVA, el precio con IVA es el siguiente

$$\text{Coste Total Materiales VCA} + \text{IVA} = 22.557 * 1.21 = 27.29\text{€}$$

VCF

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
PCB	1	2 Layer	100x20mm	AIIPCB	1.5
PANEL	1	Acryl	128x30mm	AIIPCB	3.00
KNOBS	4	1900h	https://www.thonk.co.uk/shop/1900h-d/	Black - Glow Pointer (D-Shaft)	2.48
RV1 RV2 RV3	3	10k	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	Alpha Potentiometer	4.8
U2	1	V2164	SOIC-16_3.9x9.9mm_Pitch1.27mm	VCA IC	3.27
J2 J3 J4 J5 J6 J7	6	JACK	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	2.7
C11 C12	2	47p	VJ1206A470JXQPW1 BC	Unpolarised Capacitor	0.17
C14 C17	2	220p	VJ1206Y221KXACW1 BC	Unpolarised Capacitor	0.17
C15	1	220n	VJ1206Y224KXJTW1 BC	Unpolarised Capacitor	0.085
C1 C13 C16 C2	4	560p	VJ1206Y561KXAPW1 BC	Unpolarised Capacitor	0.34
C3 C4	2	47u	VE-470M1CTR-0505	Polarised Capacitor	0.324
C10 C5 C6 C7 C8 C9	6	0.1u	C1206C104K5RACTU	Unpolarised Capacitor	0.51
D1 D2 D4 D5	4	D	1N4148W-7-F	Diode	0.58
D3	1	LM4040-5V	LM4040DYM3-5.0-TR	Voltage Reference Diode	0.298
J1	1	Conn_02x05_Odd_Even	M20-9990546	Generic connector 02x05	0.34
Q1	1	MMBT3906	MMBT3906LT3G	Transistor PNP	0.094
R1 R14 R15 R18 R4 R7 R8 R9 R22	9	100k	RC1206FR-07100KL	Resistor	0.765
R10	1	15k	RC1206FR-0715KL	Resistor	0.085
R11 R12 R13	3	39k	RC1206FR-0739KL	Resistor	0.255
R19 R26	2	510	RC1206FR-07510L	Resistor	0.17
R2 R6	2	47k	RC1206FR-0747KL	Resistor	0.17
R20 R25 R27	3	33k	RC1206FR-0733KL	Resistor	0.255
R16 R17 R21 R23	4	1k	RC1206FR-071KL	Resistor	0.34

Presupuesto

R3	1	62k	RC1206FR-0762KL	Resistor	0.085
R24 R5	2	16k	RC1206FR-0716KL	Resistor	0.17
RT1	1	5k	PT6KV-502A2020	Trimmer	0.486
U1	1	TL074	TL074CDR	Quad Operational Amplifiers	0.485
U4	1	LM13700	LM13700MX/NOPB	Dual Transconductance Amplifiers	0.986
U3 U5	2	TL072	TL072CDR	Dual Operational Amplifiers	1.004
				TOTAL	25.917

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Mermas y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Mermas y Roturas} = 25,92 * 0,02 = 0,52\text{€}$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Mermas y Roturas}$$

$$\text{Coste Total Materiales VCF} = 25,92 + 0,52 = 26,44\text{€}$$

Este precio no incluye IVA, el precio con IVA es el siguiente

$$\text{Coste Total Materiales VCF} + \text{IVA} = 26,44 * 1,21 = 31,99\text{€}$$

Sumando cada módulo obtenemos los siguientes valores:

VCO	LFO	ADSR	VCA	VCF
27.39	25.08	21.46	22.55	26.44
			TOTAL	TOTAL+IVA
			122.76	149.59

Coste de adquisición: 149.59€

2.2. Costes de mano de obra

Este apartado concreta los costes de mano de obra del sistema, analizando el coste de fabricación del prototipo y desarrollo.

Descripción	Profesional	Horas	Tasa (€/h)	Parcial
Planificación	Ingeniero Técnico	5	15	75
Diseño hardware	Ingeniero Técnico	368	15	5520
Diseño PCBs	Ingeniero Técnico	112	15	1680
Diseño software	Ingeniero Técnico	368	15	5520
Test del sistema	Especialista	50	10	500
Diseño de los módulos	Diseñador	300	15	4500
Inspección y ajuste	Ingeniero Técnico	10	15	150
			TOTAL	€17,945

Este es el presupuesto sin IVA del desarrollo de estos prototipos, que al ser 5 módulos independientes, elevan las horas de desarrollo sobre un único dispositivo.

Este precio no incluye IVA, el precio con IVA es el siguiente

$$\text{Costes Mano de Obra} + \text{IVA} = 17945 * 1.21 = 21713€$$

2.3. Presupuesto de ejecución material

El presupuesto de ejecución material (P.E.M.) se calculará en base a los costes según su naturaleza.

$$\text{PEM} = \text{Costes Materiales} + \text{Costes Mano de Obra} = 154.59 + 17945 = 18099.59€$$

Este precio no incluye IVA, el precio con IVA es el siguiente

$$\text{Costes Prototipo} = \text{PEM} * 1.21 = 18099.59 * 1.21 = 21900.5€$$

3. Presupuesto 50 unidades

En este apartado se obtendrá el presupuesto para una partida de 50 unidades con el fin de comprobar la viabilidad comercial del producto.

Al tratar con cantidades elevadas los costes de materiales se verán reducidos así como los costes de mano de obra.

Al presupuesto de ejecución material de las 50 unidades se le añadirá el del prototipo para amortizarlo, a su vez se aplicará un porcentaje correspondiente al beneficio industrial esperado y el IVA. Una vez obtenido el coste total se calculará el unitario.

3.1. Costes materiales del prototipo

VCO

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
C4,C2,C1	3	1uF	VJ1206Y105MXQTW1BC	Unpolarized capacitor	0.09
C5	1	1nF	VJ1206Y102MXQTW1BC	Unpolarized capacitor	0.03
C6,C7	2	47u	VE-470M1CTR-0505	Polarised capacitor	0.262
C8,C10	2	22pF	VJ1206A220KXQPW1BC	Unpolarized capacitor	0.06
C9	1	4.7uF	M865080640004	Polarised capacitor	0.131
C13,C3	2	1uF	M865060640001	Polarised capacitor	0.262
C14	1	10nF	VJ1206Y103MXQTW1BC	Unpolarized capacitor	0.03
C15,C11,C12	3	100nF	C1206C104K5RACTU	Unpolarized capacitor	0.09
D1,D2	2	D	1N4148W-7-F	Diode	0.246
D3	1	5.1V	BZT52C5V1-E3-08	Zener Diode	0.154
D4	1	3.3V	BZT52C3V3-E3-18	Zener Diode	0.154
D5	1	LED_R GB_5M M	WP154A4SUREQBFZGW	LED RGB, Common Cathode	0.951
J1	2	02x05	M20-9990546	Generic connector 02x05	0.308
J2,J3,J4,J5	4	Jack-2	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	1.2
J6	1	Conn_0 1x06	M20-9770646	Generic connector 01x06	0.191
Q1	1	MMBT3 904	MMBT3904	NPN Transistor	0.097
R2	1	100k	RC1206FR-07100KL	Resistor	0.02
R4	1	22k	RC1206FR-0722KL	Resistor	0.02
R5	1	15k	RC1206FR-0715KL	Resistor	0.02
R6	1	33k	RC1206FR-0733KL	Resistor	0.02
R8	1	6k8	RC1206FR-076K8L	Resistor	0.02
R10,R1,R17, R9,R3,R11,R 12,R13,R7	9	10k	RC1206FR-0710KL	Resistor	0.18
R14,R15,R16	3	100	RC1206FR-07100L	Resistor	0.06
RV21,RV2,R V3	3	10K	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	VERTICAL: B10K – 10K Linear - D shaft	3.66
SW1	1	SW_Pu sh	TL1105KF160Q	Push button switch	0.217

Presupuesto

U1,U3	2	TL072	TL072CDR	Dual Operational Amplifiers	0.814
U2	1	L7805	L7805CDT-TR	Voltage Regulator 5V,	0.503
U4	1	TC1262-33	TC1262-3.3VDBTR	Voltage Regulator 3.3V,	0.374
U5	1	STM32F334K8	STM32F334K8T6	Microcontroller	3.03
Y1	1	8MHz	LFXTAL003156Bulk	Two pin crystal	0.123
PCB	1	2 Layer	100x30mm	AllPCB	2
PANEL	1	PCB	128x30mm	AllPCB	2.00
KNOBS	3	1900h	https://www.thonk.co.uk/shop/1900h-d/	Black - Glow Pointer (D-Shaft)	1.35
				TOTAL	18.667

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Mermas y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Mermas y Roturas}$$

$$\text{Coste Total Materiales VCO-50 Unidades} = 18.667 * 1.02 * 50 = 952.01\text{€}$$

LFO

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
C1, C14	2	1uF	M865060640001	Polarised capacitor	0.262
C3, C4, C6	3	1uF	VJ1206Y105MXQTW1BC	Unpolarized capacitor	0.06
C7, C8	2	47u	VE-470M1CTR-0505	Polarised capacitor	0.324
C9, C11	2	22pF	VJ1206A220KXQPW1BC	Unpolarized capacitor	0.04
C10	1	4.7uF	M865080640004	Polarised capacitor	0.131
C12, C13, C16	3	100nF	C1206C104K5RACTU	Unpolarized capacitor	0.06
C15	1	10nF	VJ1206Y103MXQTW1BC	Unpolarized capacitor	0.02
D1, D2	2	D	1N4148W-7-F	Diode	0.242
D3	1	3.3V	BZT52C3V3-E3-18	Zener Diode	0.154
D4	1	LED_RGB_5MM	WP154A4SUREQBFZGW	LED RGB, Common Cathode	0.951
J1	1	02x05	M20-9990546	Generic connector 02x05	0.164
J2, J3, J4, J6	4	Jack-2	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	1
J5	1	Conn_01x06	M20-9770646	Generic connector 01x06	0.154
Q1	1	MMBT3904	MMBT3904	NPN Transistor	0.097
R1, R2, R4, R6, R9, R10, R12, R13, R17, R18	10	10k	RC1206FR-0710KL	Resistor	0.2
R3	1	15k	RC1206FR-0715KL	Resistor	0.02
R5, R8	2	5k6	RC1206FR-075K6L	Resistor	0.04
R7	1	33k	RC1206FR-0733KL	Resistor	0.02
R11	1	3k	RC1206FR-073KL	Resistor	0.02
R14, R15, R16	3	100	RC1206FR-07100L	Resistor	0.06
R19	1	10M	RC1206FR-0710ML	Resistor	0.02
RV1, RV2	2	10K	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	VERTICAL: B10K – 10K Linear - D shaft	2.44
SW1, SW2	2	SW_Push	TL1105KF160Q	Push button switch	0.434
U1	1	L7805	L7805CDT-TR	Voltage Regulator 5V,	0.503
U2	1	TL074	TL074CDR	Quad Operational Amplifiers	0.391
U3	1	TC1262-33	TC1262-3.3VDBTR	Voltage Regulator 3.3V,	0.374

Presupuesto

U5	1	STM32F33 4K8	STM32F334K8T6	Microcontroller	3.03
Y1	1	8MHz	LFXTAL003156Bulk	Two pin crystal	0.123
PCB	1	2 Layer	100x30mm	AIIPCB	2
PANEL	1	PCB	128x30mm	AIIPCB	2.00
KNOBS	3	1900h	https://www.thonk.co.uk/show/1900h-d/	Black - Glow Pointer (D-Shaft)	1.35
				TOTAL	16.68

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Mermas y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Mermas y Roturas}$$

$$\text{Coste Total Materiales LFO-50 Unidades} = 16.68 * 1.02 * 50 = 850.68\text{€}$$

ADSR

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
PCB	1	2 Layer	100x20mm	AllPCB	1
PANEL	1	Acryl	128x30mm	AllPCB	2.00
KNOBS	4	1900h	https://www.thonk.co.uk/shop/1900h-d/	Black - Glow Pointer (D-Shaft)	1.8
RV1 RV2 RV3 RV4	4	10k	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	Alpha Potentiometer	4.88
J3 J4 J5	3	Audio-Jack-2	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	0.75
C1 C2	2	47u	VE-470M1CTR-0505	Polarised Capacitor	0.262
C3 C4 C6	3	0.1uF	C1206C104K5RACTU	Unpolarised Capacitor	0.06
C7 C8	2	22p	VJ1206A220KXQPW1BC	Unpolarised Capacitor	0.04
D1 D2	2	D	1N4148W-7-F	Diode	0.246
D3	1	LED	LEDs:LED_D3.0mm	LED Diode	0.085
J1	1	Conn_02 x05_Odd _Even	M20-9990546	Generic connector 02x05	0.154
J2	1	Conn_01 x08	M20-9990534	Generic connector 01x08	0.184
Q1 Q2	2	Q_NPN_ BEC	MMBT3904	Transistor NPN	0.194
R1	1	100	RC1206FR-07100L	Resistor	0.02
R10 R11 R12 R13 R14 R2 R3 R4 R5 R6 R9	11	10k	RC1206FR-07100KL	Resistor	0.22
R7	1	6k	RC1206FR-076KL	Resistor	0.02
R8	1	0	RC1206FR-070L	Resistor	0.02
U1	1	MCP472 5	MCP4725	DAC Converter	0.695
U2	1	L7805	L7805CDT-TR	Linear Regulator	0.503
U3	1	TL072	TL072CDR	Dual Operational Amplifiers	0.502
U4	1	ATMEGA 328P-AU	ATMEGA328P-AU	Microcontroller	1.61
Y1	1	Crystal	LFXTAL003156Bulk	Crystal	0.123
				TOTAL	15.368

Presupuesto

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Mermas y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Mermas y Roturas}$$

$$\text{Coste Total Materiales ADSR-50 Unidades} = 15.368 * 1.02 * 50 = 783.78\text{€}$$

VCA

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
PCB	1	2 Layer	100x30mm	AllPCB	2
PANEL	1	Acryl	128x30mm	AllPCB	2.00
KNOBS	2	1900h	https://www.thonk.co.uk/shop/1900h-d/	Black - Glow Pointer (D-Shaft)	0.9
RV1 RV3	2	100k	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	Alpha Potentiometer	2.44
RV2 RV4	2	100k	https://www.thonk.co.uk/shop/ttpots/	Song Huei Potentiometer	1.7
J1 J2 J4 J5 J6 J7 J8 J9	8	Jack-2	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	2
C1 C2 C3 C4	4	0.1uF	C1206C104K5RAC TU	Unpolarized capacitor	0.36
D1 D2	2	D	1N4148W-7-F	Diode	0.246
D3 D4	2	LED	BZT52C5V1-E3-08	LED Diode	0.16
J3	2	02x05	M20-9990546	Generic connector 02x05	0.328
R1 R12 R15 R18 R19 R21 R4 R5 R6 R8	10	100k	RC1206FR- 07100KL	Resistor	0.2
R10 R23	2	220k	RC1206FR- 07220KL	Resistor	0.04
R11 R24	2	30k	RC1206FR- 0730KL	Resistor	0.04
R16 R25	2	10k	RC1206FR- 0710KL	Resistor	0.04
R17 R26	2	1k	RC1206FR-071KL	Resistor	0.04
R13 R14 R2 R20 R3 R7	6	22k	RC1206FR- 0722KL	Resistor	0.12
R22 R9	2	220	RC1206FR-07220L	Resistor	0.04
U1 U3	2	TL072	TL072CDR	Dual Operational Amplifiers	0.808
U2	1	LM13700	LM13700MX/NOP B	Dual Transconductance Amplifiers	0.809
				TOTAL	14.271

Presupuesto

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Mermas y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Mermas y Roturas}$$

$$\text{Coste Total Materiales VCA-50 Unidades} = 14.271 * 1.02 * 50 = 727.82\text{€}$$

VCF

REFERENCE	QUANTITY	VALUE	REF VENDOR	DESCRIPTION	PRICE
PCB	1	2 Layer	100x20mm	AIIPCB	1
PANEL	1	Acryl	128x30mm	AIIPCB	2.00
KNOBS	4	1900h	https://www.thonk.co.uk/shop/1900h-d/	Black - Glow Pointer (D-Shaft)	1.8
RV1 RV2 RV3	3	10k	https://www.thonk.co.uk/shop/alpha-9mm-pots-dshaft/	Alpha Potentiometer	3.66
U2	1	V2164	SOIC-16_3.9x9.9mm_Pitch1.27mm	VCA IC	3
J2 J3 J4 J5 J6 J7	6	JACK	https://www.thonk.co.uk/shop/3-5mm-jacks/	Thonkiconn Jacks (PJ301M-12)	1.5
C11 C12	2	47p	VJ1206A470JXQPW1 BC	Unpolarised Capacitor	0.04
C14 C17	2	220p	VJ1206Y221KXACW1 BC	Unpolarised Capacitor	0.04
C15	1	220n	VJ1206Y224KXJTW1 BC	Unpolarised Capacitor	0.02
C1 C13 C16 C2	4	560p	VJ1206Y561KXAPW1 BC	Unpolarised Capacitor	0.08
C3 C4	2	47u	VE-470M1CTR-0505	Polarised Capacitor	0.262
C10 C5 C6 C7 C8 C9	6	0.1u	C1206C104K5RACTU	Unpolarised Capacitor	0.12
D1 D2 D4 D5	4	D	1N4148W-7-F	Diode	0.492
D3	1	LM4040-5V	LM4040DYM3-5.0-TR	Voltage Reference Diode	0.209
J1	1	Conn_02x05_Odd_Even	M20-9990546	Generic connector 02x05	0.34
Q1	1	MMBT3906	MMBT3906LT3G	Transistor PNP	0.094
R1 R14 R15 R18 R4 R7 R8 R9 R22	9	100k	RC1206FR-07100KL	Resistor	0.18
R10	1	15k	RC1206FR-0715KL	Resistor	0.02
R11 R12 R13	3	39k	RC1206FR-0739KL	Resistor	0.06
R19 R26	2	510	RC1206FR-07510L	Resistor	0.04
R2 R6	2	47k	RC1206FR-0747KL	Resistor	0.04
R20 R25 R27	3	33k	RC1206FR-0733KL	Resistor	0.06
R16 R17 R21 R23	4	1k	RC1206FR-071KL	Resistor	0.08

Presupuesto

R3	1	62k	RC1206FR-0762KL	Resistor	0.02
R24 R5	2	16k	RC1206FR-0716KL	Resistor	0.04
RT1	1	5k	PT6KV-502A2020	Trimmer	0.406
U1	1	TL074	TL074CDR	Quad Operational Amplifiers	0.405
U4	1	LM13700	LM13700MX/NOPB	Dual Transconductance Amplifiers	0.809
U3 U5	2	TL072	TL072CDR	Dual Operational Amplifiers	1.004
				TOTAL	17.821

Al coste de materiales se le añadirá un porcentaje del 2% para cubrir variaciones causadas por mermas y roturas, así pues el coste total será:

$$\text{Coste Merms y Roturas} = \text{Coste Materiales} * 2\%$$

$$\text{Coste Total Materiales} = \text{Coste Materiales} + \text{Coste Merms y Roturas}$$

$$\text{Coste Total Materiales VCF-50 Unidades} = 17.821 * 1.02 * 50 = 908.87\text{€}$$

Sumando los costes de materiales de cada módulo obtenemos los siguientes valores:

VCO	LFO	ADSR	VCA	VCF
952.01	850.68	783.78	727.82	908.97
				TOTAL
				4223.26

Coste de adquisición de materiales: 4223.26€

3.2. Costes de mano de obra

Los costes de mano de obra para esta partida de 50 unidades de cada módulo se basan en el montaje de los materiales adquiridos, la programación de los dispositivos, el testeo y su packaging.

Descripción	Profesional	Horas	Tasa (€/h)	Parcial
Montaje PCB	Técnico	500	10	5000
Programación módulos	Ingeniero Técnico	10	15	150
Testeo módulos	Ingeniero Técnico	20	15	300
Packaging	Técnico	10	10	100
			TOTAL	€5,550

Coste de mano de obra: 5550€

3.3. Presupuesto de ejecución material

El presupuesto de ejecución material (P.E.M.) se calculará en base a los costes según su naturaleza.

$$PEM = Costes Materiales + Costes Mano de Obra + PEM Prototipo$$

$$PEM = 4223.26 + 5550 + 21900.5 = 31673.76€$$

3.4. Beneficio industrial

El beneficio industrial es el porcentaje de ganancia que se obtiene. Se calcula añadiendo un porcentaje al presupuesto de ejecución material. Su valor se establecerá en este documento en un 15%.

$$Beneficio Industrial = PEM * 25\% = 31673.76 * 0.25 = 7918.44€$$

3.5. Coste total 50 unidades

Para el calculo final del coste total del proyecto se sumará el beneficio industrial al presupuesto de ejecución material. Al valor obtenido se le aplicará un incremento del 21% correspondiente al IVA.

$$Coste Total 50 = PEM + Beneficio Industrial = 31673.76 + 7918.44€ = 39592.2€$$

$$Coste Total 50 IVA = (PEM + Beneficio Industrial) * 1.21 = 47906.5€$$

3.6. Precio de venta unitario

Precio de venta del dispositivo consistente en 5 módulos (VCO, LFO, ADSR, VCF, VCA)

$$\text{Precio de Venta} = \frac{\text{Coste total 50 IVA}}{50} = 958.13€$$

3.7. Posición en el mercado

Con el precio de venta ya decidido, es importante conocer nuestra posición en el mercado frente a los competidores. Cabe recordar que por nuestra parte se ofrece un sistema de síntesis básico, por tanto se va a comparar con los productos más similares a este.

Dreadbox Abyss: Es un sintetizador analógico de 4 voces, incluye todos los componentes presentes en nuestro sistema: VCO, LFO, VCA, VCF y ADSR. Este sistema es similar al desarrollado al ser un sintetizador básico como el nuestro. En la Figura 1 podemos observar su apariencia:



Figura 1: Sintetizador Dreadbox Abyss

Su precio en tienda es de **1083€**, una cifra similar a lo que se vendería nuestro sistema.

Endorphin.es Shuttle System: Es un sintetizador analógico similar en cuanto a módulos y funcionalidad, ya que incluye todos los componentes presentes en nuestro sistema: VCO, LFO, VCA, VCF y ADSR y más módulos adicionales. En la Figura 2 podemos observar su apariencia:



Figura 2: Sintetizador Endorphin.es Shuttle System

Su precio en tienda es de 2222€, una cifra que supera más del doble la nuestra, pero está debidamente justificado, ya que incluye módulos y funcionalidades que nuestro sistema no tiene.

Pittsburgh Modular Lifeforms System 301: Es un sintetizador analógico similar en cuanto a módulos y funcionalidad, ya que incluye todos los componentes presentes en nuestro sistema: VCO, LFO, VCA, VCF y ADSR y más modulos adicionales. En la Figura 3 podemos observar su apariencia:



Figura 3: Sintetizador Pittsburgh Modular Lifeforms System 301

Su precio en tienda es de 1839, una cifra que es considerablemente más alta que la nuestra, pero está debidamente justificado, ya que incluye módulos y funcionalidades que nuestro sistema no tiene, concretamente el controlador incluido en el sistema, que permite no tener que utilizar un controlador externo.

La **conclusión** de este posicionamiento en el mercado, es que nuestro sistema se posiciona en el mercado de los sintetizadores completos, preparados para conectar y sonar. Estos sintetizadores son sistemas complejos, bien acabados y con materiales

de alta calidad, por tanto nuestra posición en el mercado es esta, los precios son altos pero están justificados al ser sistemas complejos de buena calidad.

Frente a nuestros competidores, nuestro precio es menor debido a que lo que se ofrece es un sistema básico, frente a sus productos que tienen más funcionalidades: Por tanto, podríamos decir que está igualado el coste por funcionalidad.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño e implementación de un sintetizador de audio modular basado en síntesis substractiva

PLIEGO DE CONDICIONES - TFG

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Gutiérrez-Ravé Olmos, Jorge

Tutor: Coll Arnau, Salvador

ÍNDICE:

1. Definición y alcance del pliego de condiciones	3
2. Condiciones y normas de carácter general	5
3. Condiciones técnicas	7
3.1 Condiciones técnicas de los materiales	9
3.1.1 Placa circuito impreso	9
3.1.2 Circuitos integrados	9
3.1.3 Condensadores	11
3.1.4 Resistencias	11
3.1.5 Osciladores	11
3.1.6 LED	12
3.1.7 Panel	12
3.2 Programación del dispositivo	13
3.3 Pruebas y ajustes finales	13
3.3.1 Prueba de hardware	13
3.3.2 Prueba de funcionamiento	13
4. Condiciones facultativas	15
4.1 Obligaciones de la propiedad	15
4.2 Obligaciones y derechos del contratista	15
4.3 Obligaciones y facultades de la dirección facultativa	15

1. Definición y alcance del pliego de condiciones

El objeto de este documento es fijar las condiciones técnicas mínimas que debe cumplir el sistema de sintetizador de audio modular basado en síntesis substractiva.

El ámbito de aplicación de este documento se extiende a los sistemas electrónicos que forman parte del diseño.

Se precisarán las especificaciones que deben cumplir los componentes electrónicos incluidos en el circuito del dispositivo. También se detallan las normas y reglamentaciones legales de cada uno de estos componentes.

Podrán ampliarse los conocimientos sobre los componentes utilizados mediante consulta del documento de planos donde se incluyen las hojas de características. Se establecerán las condiciones facultativas mínimas entre las distintas partes participantes del proyecto.

2. Condiciones y normas de carácter general

En este documento se precisan los componentes electrónicos utilizados en el circuito del sistema. En determinados casos podrán sustituirse estos elementos por soluciones diferentes a las exigidas en el pliego siempre que se justifique su necesidad y no implique rebajar el nivel de rendimiento del dispositivo.

En caso de fallos en el funcionamiento tras realizar dichos cambios, la responsabilidad recaerá en el agente que decidiera realizarlos. Se consideran precisas las características fijadas por los fabricantes, por lo que no se realizarán medidas auxiliares para comprobar que los componentes cumplan dichas especificaciones.

3. Condiciones técnicas

3.1 Condiciones técnicas de los materiales

Se definirán los materiales que se utilizarán en el proyecto para la materialización y puesta a punto del hardware y las pruebas necesarias para la puesta a punto del dispositivo.

Se hará referencia a normativas y reglamentos nacionales o internacionales a los que están sujetos cada uno de los componentes que conforman el sistema electrónico.

3.1.1 Placa circuito impreso

Este apartado es aplicable a todos los módulos del proyecto (VCO, LFO, ADSR, VCA, VCF). Para la fabricación de la placa se recurrirá a la subcontratación de una empresa externa a la que se le enviará un archivo con la información pertinente al diseño de la placa PCB. Las medidas finales de la placa de circuito impreso serán de 100mm x 30mm.

La **normativa** aplicable a los circuitos impresos será:

- UNE-EN 123000/A1:1995 Especificación genérica: Placas de circuitos impresos.
- UNE-EN 60249:1997 Materiales base para circuitos impresos.
- UNE-EN 61249:2005 Materiales para estructuras de interconexión.
- UNE-EN 61191-1:2013 Conjuntos de tarjetas con circuito impreso, especificación genérica. Conjuntos eléctricos y electrónicos soldados utilizando tecnología de montaje en superficie y tecnología de montaje asociado.
- UNE-EN 60194:2006 circuitos impresos, manufactura y ensamblaje, términos y definiciones.
- UNE-EN 61188-5-1:2002 Circuitos impresos, diseño y uso.
- UNE-EN 123300:1992 Especificación intermedia: placas de circuito multicapa.
- UNE-EN 160100:1997 Especificación intermedia: Aprobación de la capacidad de fabricantes de conjuntos de tarjetas de circuito impreso de calidad asegurada.

3.1.2 Circuitos integrados

Los encapsulados que se utilizarán en el dispositivo **VCO** serán los indicados en este apartado:

- **L7805**: Empaquetado TO-252-2.
- **TC1262-33**: Empaquetado TO-223-3
- **STM32F334K8**: Empaquetado LQFP-32
- **TL072**: Empaquetado SOIC-8

L7805, **TC1262-33** y **TL072** pueden ser sustituidos por componentes con similares características si fuese necesario. No obstante los circuitos integrados restantes no deben ser sustituidos, de hacerlo, se compromete el funcionamiento del dispositivo. Los encapsulados que se utilizarán en el dispositivo **LFO** serán los indicados en este apartado:

- **L7805**: Empaquetado TO-252-2.
- **TC1262-33**: Empaquetado TO-223-3
- **STM32F334K8**: Empaquetado LQFP-32
- **TL074**: Empaquetado SOIC-14

L7805, **TC1262-33** y **TL074** pueden ser sustituidos por componentes con similares características si fuese necesario. No obstante los circuitos integrados restantes no deben ser sustituidos, de hacerlo, se compromete el funcionamiento del dispositivo.

Los encapsulados que se utilizarán en el dispositivo **ADSR** serán los indicados en este apartado:

- **L7805**: Empaquetado TO-252-2.
- **MCP4725**: Empaquetado SOT-23-6
- **ATMEGA328AU**: Empaquetado TQFP-32
- **TL072**: Empaquetado SOIC-8

L7805, y **TL072** pueden ser sustituidos por componentes con similares características si fuese necesario. No obstante los circuitos integrados restantes no deben ser sustituidos, de hacerlo, se compromete el funcionamiento del dispositivo.

Los encapsulados que se utilizarán en el dispositivo **VCA** serán los indicados en este apartado:

- **TL072**: Empaquetado SOIC-8
- **LM13700**: Empaquetado SOIC-16

TL072 puede ser sustituido por componentes con similares características si fuese necesario. No obstante los circuitos integrados restantes no deben ser sustituidos, de hacerlo, se compromete el funcionamiento del dispositivo.

Los encapsulados que se utilizarán en el dispositivo **VCF** serán los indicados en este apartado:

- **TL072**: Empaquetado SOIC-8
- **TL074**: Empaquetado SOIC-14
- **LM13700**: Empaquetado SOIC-16

TL072 y **TL074** pueden ser sustituidos por componentes con similares características si fuese necesario. No obstante los circuitos integrados restantes no deben ser sustituidos, de hacerlo, se compromete el funcionamiento del dispositivo.

Se tomarán precauciones cuando se manipulen los componentes aquí listados para evitar que posible estática dañe su estructura. Los encapsulados se colocarán con las muescas orientadas tal y como se indica en los planos.

3.1.3 Condensadores

Este apartado es aplicable a todos los módulos del proyecto (VCO, LFO, ADSR, VCA, VCF). Los condensadores que se utilizarán serán de tamaño reducido con encapsulado de montaje superficial 1206" (3216 metric) y con tolerancias iguales o inferiores al 10%.

La **normativa** aplicable a los condensadores es la siguiente:

- UNE-EN 130200:1992 Condensadores fijos de tantalio con electrolítico sólido o no sólido.
- UNE-EN 60062:2005 CORR:2007 Códigos para el marcado de resistencias y de condensadores.
- UNE20547:1974 Código para la fecha de componentes electrónicos.
- UNE 20543:1985 Condensadores utilizados en los equipos electrónicos.
- UNE-EN 62490-2:2010 Condensadores montados en superficie para uso en equipos electrónicos.

3.1.4 Resistencias

Este apartado es aplicable a todos los módulos del proyecto (VCO, LFO, ADSR, VCA, VCF). Se usarán resistencias con encapsulado de montaje superficial 1206" (3216 metric) con tolerancias inferiores al 5%. Las resistencias cumplirán la siguiente **normativa**:

- UNE-EN 60062:2005 CORR:2007 Códigos para el marcado de resistencias y de condensadores.
- UNE-EN 60115-9-1:2004 Resistencias fijas para uso en equipos electrónicos.
- UNE-EN 140401:2009 Resistencias fijas de montaje sobre superficie no bobinada de baja disipación.
- UNE-EN 140401-804:2011/A1:2013 Resistencias SMD fijas de película de baja potencia y alta estabilidad. Rectangular.

3.1.5 Osciladores

Este apartado es aplicable a los módulos digitales del proyecto (VCO, LFO, ADSR). Se emplearán osciladores con encapsulado de montaje superficial. Siendo Q1 de 5 x 10mm a 8MHz. La **normativa** que deberán cumplir se listará a continuación:

- UNE-EN 60679-1:2007 Osciladores controlados por cristal de cuarzo con aseguramiento de calidad. Especificación genérica
- UNE-EN 60679-3:2013 Osciladores controlados por cristal de cuarzo con aseguramiento de calidad. Contornos y conexiones de plomo normalizados.

3.1.6 LED

Este apartado es aplicable a los módulos digitales del proyecto (VCO, LFO, ADSR). Se emplearán LEDs RGB con encapsulado de montaje en panel y una corriente directa de 20mA. En el caso de los módulos analógicos (VCA, VCF) se emplearán LEDs de propósito general de color azul, con encapsulado de montaje en panel y una corriente directa de 15mA.

3.1.7 Panel

Este apartado es aplicable a todos los módulos del proyecto (VCO, LFO, ADSR, VCA, VCF). El panel será realizado en circuito impreso, por tanto se le aplican todas las condiciones y normativas comentadas anteriormente en el apartado 3.1.1 Placa circuito impreso.

3.2 Programación del dispositivo

Este apartado es aplicable a los módulos digitales del proyecto (VCO, LFO, ADSR). El dispositivo electrónico se programará con el uso de una herramienta STLink V2, del fabricante STMicroElectronics. Para ello se procederá a alimentar el módulo con una fuente de alimentación compatible con el estándar Eurorack, entonces desde el computador se flashearé el dispositivo con el archivo hexadecimal del proyecto.

3.3 Pruebas y ajustes finales

Para garantizar el correcto funcionamiento del dispositivo electrónico, se realizarán una serie de pruebas sobre las funciones básicas del sistema. Este apartado es aplicable a todos los módulos del proyecto (VCO, LFO, ADSR, VCA, VCF).

3.3.1 Prueba de hardware

Se llevará a cabo esta prueba para comprobar que la electrónica funciona según lo esperado. Se conectará el módulo a la fuente de alimentación y se buscarán cortocircuitos, en caso de no haberlos, habrá superado esta prueba.

3.3.2 Prueba de funcionamiento

Esta prueba se lleva a cabo con las mismas condiciones que la prueba de hardware, pero en este caso se debe visualizar la señal de salida del dispositivo con un osciloscopio. En caso de que la señal sea correcta, se procederá a comprobar el funcionamiento de sus potenciómetros, botones, entradas y salidas.

En caso de que todo muestre un comportamiento adecuado, el módulo estaría preparado para el embalaje y su posterior distribución.

4. Condiciones facultativas

Se describirán los derechos y obligaciones de las partes presentes en el desarrollo y ejecución del proyecto, siendo estos la contrata la propiedad y la dirección facultativa. Este apartado es aplicable a todos los módulos del proyecto (VCO, LFO, ADSR, VCA, VCF).

4.1 Obligaciones de la propiedad

Será contrata cualquier persona o entidad, física o jurídica que impulse, programe o financie con recursos propios o ajenos, la ejecución del proyecto.

Sus obligaciones son:

Facilitar la documentación e información previa necesarias para la redacción del proyecto. Gestionar y obtener las preceptivas licencias y autorizaciones administrativas.

4.2 Obligaciones y derechos del contratista

Se denominará contratista a la persona física o entidad, pública o privada, encargada de la ejecución y puesta en marcha del proyecto.

Son obligaciones del proyectista:

- Redactar el proyecto con sujeción a la normativa vigente y a lo que se haya establecido en el contrato.
- Conocer el proyecto en todas sus partes.
- Otorgar accesibilidad habilitando un local para consulta del proyecto y localización de los responsables de su ejecución.
- Ejecución de trabajos no especificados necesarios para llevar a cabo el proyecto y su correspondiente documentación.
- Obligación a seguir las indicaciones del proyecto y la dirección facultativa.
- Notificar la inicialización, finalización realización de pruebas o certificaciones del proyecto o de alguna de sus partes, así como establecer protocolos a seguir en los actos mencionados.

4.3 Obligaciones y facultades de la dirección facultativa

Será dirección facultativa el representante de la propiedad, que velará por los intereses de esta ante el ejecutor del proyecto.

Sus **obligaciones** serán:

- Supervisar los aspectos del proyecto que puedan comprometer su fiabilidad, calidad y seguridad durante su ejecución. Cumplimentar los certificados de dirección exigidos por la normativa vigente.

- Presencia en los momentos de desarrollo del proyecto que se convenga.
- Responsabilidad derivada de ser la autoridad técnica en materia de interpretación de las indicaciones dadas en el proyecto.
- Realizar un seguimiento para procurar que se cumplan los plazos.
- Informar a la propiedad de la marcha del proyecto y de las contingencias que surjan.
- Aportar soluciones técnicas a problemas imprevistos durante la ejecución.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño e implementación de un sintetizador de audio modular basado en síntesis substractiva

ANEXO CÓDIGO - TFG

Grado en Ingeniería Electrónica Industrial y Automática

Autor: Gutiérrez-Ravé Olmos, Jorge

Tutor: Coll Arnau, Salvador

ÍNDICE:

1. Introducción	4
2. Librerías requeridas	7
2.1 Librerías propias	7
Ade_nano.h:	7
Dac_nano.h:	14
Dma_nano.h:	17
Exti_nano.h:	19
Gpio_nano.h:	20
Pwm_nano.h:	23
Tim_nano.h:	25
2.1 Librerías externas	29
Wire.h:	29
Adafruit_MCP4725.h:	35
3. Código de los módulos	37
3.1. VCO	37
3.2. LFO	43
3.3. ADSR	57

1. Introducción

En este documento muestra el código necesario para poner en funcionamiento los módulos del sistema de síntesis sustractivo de audio. Este código incluye tanto las librerías necesarias como el código propio de estos módulos.

Los módulos correspondientes a este código son los siguientes:

VCO: Su función es generar una señal oscilante de una frecuencia comprendida entre [20-1000]Hz y de amplitud [-5,5]V, ajustable en frecuencia y forma de onda. Para lograr este cometido, vamos a integrar el **STM32F334K8**.

LFO: Genera una señal oscilante de una frecuencia comprendida entre [0.01-100]Hz de amplitud en la salida unipolar [0,5]V y en la bipolar [-2.5,2.5]V ajustable en frecuencia y forma de onda. Para lograr este cometido, vamos a integrar el **STM32F334K8**.

ADSR: Este módulo se emplea para dar expresividad a los sonidos, permitiendo ajustar la forma de un control de voltaje respecto al tiempo. Esto hace que las señales de control de voltaje adquieran una forma más dinámica y sobre la cual podremos ajustar su avance respecto al tiempo. Este ajuste tiene 4 partes progresivas: **Attack, Decay, Sustain, Release**. Para lograr este cometido, vamos a integrar el **ATMega328**

2. Librerías requeridas

2.1 Librerías propias

Con el dispositivo hardware ya en funcionamiento, el primer paso en cuanto al software es crear librerías del proyecto para facilitar la escritura del código, esto es un proceso costoso pero que si se realiza de la forma correcta, solo se tendrá que hacer una vez, mientras que se utilizará muchas.

Adc_nano.h:

Librería escrita para el manejo del conversor analógico digital (ADC), su inicialización y procesado.

```
#include "adc_nano.h"

void configureADC(void)
{
    __IO uint32_t wait_loop_index = 0;

    /*## Configuration of GPIO used by ADC channels #####*/

    /* Note: On this STM32 device, ADC1 channel 2 is mapped on GPIO pin PA.01 */

    /* Enable GPIO Clock */
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOA);

    /* Configure GPIO in analog mode to be used as ADC input */
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_0, LL_GPIO_MODE_ANALOG);
    /* Configure GPIO in analog mode to be used as ADC input */
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_1, LL_GPIO_MODE_ANALOG);
    /* Configure GPIO in analog mode to be used as ADC input */
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_2, LL_GPIO_MODE_ANALOG);

    /*## Configuration of NVIC #####*/
    /* Configure NVIC to enable ADC1 interruptions */
    NVIC_SetPriority(ADC1_2_IRQn, 0); /* ADC IRQ greater priority than DMA IRQ */
    NVIC_EnableIRQ(ADC1_2_IRQn);

    /*## Configuration of ADC #####*/

    /*## Configuration of ADC hierarchical scope: common to several ADC #####*/

    /* Enable ADC clock (core clock) */
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_ADC12);

    /* Note: Hardware constraint (refer to description of the functions          */
    /*       below):                                                                */
    /*       On this STM32 serie, setting of these features is conditioned to    */
    /*       ADC state:                                                            */
}
```

```

/*      All ADC instances of the ADC common group must be disabled.      */
/* Note: In this example, all these checks are not necessary but are      */
/*      implemented anyway to show the best practice usages                */
/*      corresponding to reference manual procedure.                       */
/*      Software can be optimized by removing some of these checks, if    */
/*      they are not relevant considering previous settings and actions    */
/*      in user application.                                              */
if(__LL_ADC_IS_ENABLED_ALL_COMMON_INSTANCE() == 0)
{
    /* Note: Call of the functions below are commented because they are    */
    /*      useless in this example:                                        */
    /*      setting corresponding to default configuration from reset state.*/

    /* Set ADC clock (conversion clock) common to several ADC instances */
    LL_ADC_SetCommonClock(__LL_ADC_COMMON_INSTANCE(ADC1), LL_ADC_CLOCK_SYNC_PCLK_DIV2);

    /* Set ADC measurement path to internal channels */
    LL_ADC_SetCommonPathInternalCh(__LL_ADC_COMMON_INSTANCE(ADC1),
(LL_ADC_PATH_INTERNAL_VREFINT | LL_ADC_PATH_INTERNAL_TEMPSENSOR));

    /* Delay for ADC temperature sensor stabilization time.                */
    /* Compute number of CPU cycles to wait for, from delay in us.        */
    /* Note: Variable divided by 2 to compensate partially                  */
    /*      CPU processing cycles (depends on compilation optimization).    */
    /* Note: If system core clock frequency is below 200kHz, wait time    */
    /*      is only a few CPU processing cycles.                            */
    /* Note: This delay is implemented here for the purpose in this example.*/
    /*      It can be optimized if merged with other delays                */
    /*      during ADC activation or if other actions are performed        */
    /*      in the meantime.                                               */
    wait_loop_index = ((LL_ADC_DELAY_TEMPSENSOR_STAB_US * (SystemCoreClock / (100000 * 2)))
/ 10);
    while(wait_loop_index != 0)
    {
        wait_loop_index--;
    }

    /*## Configuration of ADC hierarchical scope: multimode #####*/

    /* Set ADC multimode configuration */
    // LL_ADC_SetMultimode(__LL_ADC_COMMON_INSTANCE(ADC1), LL_ADC_MULTI_INDEPENDENT);

    /* Set ADC multimode DMA transfer */
    // LL_ADC_SetMultiDMATransfer(__LL_ADC_COMMON_INSTANCE(ADC1),
LL_ADC_MULTI_REG_DMA_EACH_ADC);

    /* Set ADC multimode: delay between 2 sampling phases */
    // LL_ADC_SetMultiTwoSamplingDelay(__LL_ADC_COMMON_INSTANCE(ADC1),
LL_ADC_MULTI_TWOSMP_DELAY_1CYCLE);

}

/*## Configuration of ADC hierarchical scope: ADC instance #####*/

/* Note: Hardware constraint (refer to description of the functions      */
/*      below):                                                            */

```

Anexo Código

```
/*      On this STM32 serie, setting of these features is conditioned to */
/*      ADC state: */
/*      ADC must be disabled. */

/### Configuration of ADC hierarchical scope: ADC group regular #####*/

/* Note: Hardware constraint (refer to description of the functions */
/*      below): */
/*      On this STM32 serie, setting of these features is conditioned to */
/*      ADC state: */
/*      ADC must be disabled or enabled without conversion on going */
/*      on group regular. */
if ((LL_ADC_IsEnabled(ADC1) == 0) ||
    (LL_ADC_REG_IsConversionOngoing(ADC1) == 0) )
{
    /* Set ADC group regular trigger source */
    LL_ADC_REG_SetTriggerSource(ADC1, LL_ADC_REG_TRIG_SOFTWARE);

    /* Set ADC group regular trigger polarity */
    // LL_ADC_REG_SetTriggerEdge(ADC1, LL_ADC_REG_TRIG_EXT_RISING);

    /* Set ADC group regular continuous mode */
    LL_ADC_REG_SetContinuousMode(ADC1, LL_ADC_REG_CONV_SINGLE);

    /* Set ADC group regular conversion data transfer */
    LL_ADC_REG_SetDMATransfer(ADC1, LL_ADC_REG_DMA_TRANSFER_UNLIMITED);

    /* Set ADC group regular overrun behavior */
    LL_ADC_REG_SetOverrun(ADC1, LL_ADC_REG_OVR_DATA_OVERWRITTEN);

    /* Set ADC group regular sequencer */
    /* Note: On this STM32 serie, ADC group regular sequencer is */
    /*      fully configurable: sequencer length and each rank */
    /*      affectation to a channel are configurable. */
    /*      Refer to description of function */
    /*      "LL_ADC_REG_SetSequencerLength()". */

    /* Set ADC group regular sequencer length and scan direction */
    LL_ADC_REG_SetSequencerLength(ADC1, LL_ADC_REG_SEQ_SCAN_ENABLE_3RANKS);

    /* Set ADC group regular sequencer discontinuous mode */
    // LL_ADC_REG_SetSequencerDiscont(ADC1, LL_ADC_REG_SEQ_DISCONT_DISABLE);

    /* Set ADC group regular sequence: channel on the selected sequence rank. */
    LL_ADC_REG_SetSequencerRanks(ADC1, LL_ADC_REG_RANK_1, LL_ADC_CHANNEL_1);
    LL_ADC_REG_SetSequencerRanks(ADC1, LL_ADC_REG_RANK_2, LL_ADC_CHANNEL_2);
    LL_ADC_REG_SetSequencerRanks(ADC1, LL_ADC_REG_RANK_3, LL_ADC_CHANNEL_3);
}

/### Configuration of ADC hierarchical scope: channels #####*/

/* Note: Hardware constraint (refer to description of the functions */
/*      below): */
/*      On this STM32 serie, setting of these features is conditioned to */
/*      ADC state: */
/*      ADC must be disabled or enabled without conversion on going */
```

```

/*      on either groups regular or injected.      */
if ((LL_ADC_IsEnabled(ADC1) == 0)                ||
    ((LL_ADC_REG_IsConversionOngoing(ADC1) == 0) &&
     (LL_ADC_INJ_IsConversionOngoing(ADC1) == 0) ) )
{
  /* Set ADC channels sampling time */
  /* Note: Considering interruption occurring after each ADC group      */
  /*      regular sequence conversions      */
  /*      (IT from DMA transfer complete),      */
  /*      select sampling time and ADC clock with sufficient      */
  /*      duration to not create an overhead situation in IRQHandler.      */
  /* Note: Set long sampling time due to internal channels (VrefInt,      */
  /*      temperature sensor) constraints.      */
  /*      Refer to description of function      */
  /*      "LL_ADC_SetChannelSamplingTime()".      */
  LL_ADC_SetChannelSamplingTime(ADC1, LL_ADC_CHANNEL_1, LL_ADC_SAMPLINGTIME_601CYCLES_5);
  LL_ADC_SetChannelSamplingTime(ADC1, LL_ADC_CHANNEL_2, LL_ADC_SAMPLINGTIME_601CYCLES_5);
  LL_ADC_SetChannelSamplingTime(ADC1, LL_ADC_CHANNEL_3, LL_ADC_SAMPLINGTIME_601CYCLES_5);

  /* Set mode single-ended or differential input of the selected      */
  /* ADC channel.      */
  // LL_ADC_SetChannelSingleDiff(ADC1, LL_ADC_CHANNEL_2, LL_ADC_SINGLE_ENDED);
}

/*## Configuration of ADC transversal scope: analog watchdog #####*/

/* Set ADC analog watchdog channels to be monitored */
// LL_ADC_SetAnalogWDMonitChannels(ADC1, LL_ADC_AWD1, LL_ADC_AWD_DISABLE);

/* Set ADC analog watchdog thresholds */
// LL_ADC_ConfigAnalogWDThresholds(ADC1, LL_ADC_AWD1, 0xFF, 0x00);

/*## Configuration of ADC transversal scope: oversampling #####*/

/* Note: Feature not available on this STM32 serie */

/*## Configuration of ADC interruptions #####*/
/* Enable interruption ADC group regular end of sequence conversions */
LL_ADC_EnableIT_EOS(ADC1);

/* Enable interruption ADC group regular overrun */
LL_ADC_EnableIT_OVR(ADC1);

/* Note: in this example, ADC group regular end of conversions      */
/*      (number of ADC conversions defined by DMA buffer size)      */
/*      are notified by DMA transfer interruptions).      */
/*      ADC interruptions of end of conversion are enabled optionally,      */
/*      as demonstration purpose in this example.      */
}

void activateADC(void)
{
  __IO uint32_t wait_loop_index = 0;

```

Anexo Código

```
#if (USE_TIMEOUT == 1)
uint32_t Timeout = 0; /* Variable used for timeout management */
#endif /* USE_TIMEOUT */

if (LL_ADC_IsEnabled(ADC1) == 0)
{
    /* Enable ADC internal voltage regulator */
    LL_ADC_EnableInternalRegulator(ADC1);

    /* Delay for ADC internal voltage regulator stabilization. */
    /* Compute number of CPU cycles to wait for, from delay in us. */
    /* Note: Variable divided by 2 to compensate partially */
    /* CPU processing cycles (depends on compilation optimization). */
    /* Note: If system core clock frequency is below 200kHz, wait time */
    /* is only a few CPU processing cycles. */
    wait_loop_index = ((LL_ADC_DELAY_INTERNAL_REGUL_STAB_US * (SystemCoreClock / (100000 *
2))) / 10);
    while(wait_loop_index != 0)
    {
        wait_loop_index--;
    }

    /* Run ADC self calibration */
    LL_ADC_StartCalibration(ADC1, LL_ADC_SINGLE_ENDED);

    /* Poll for ADC effectively calibrated */
    #if (USE_TIMEOUT == 1)
    Timeout = ADC_CALIBRATION_TIMEOUT_MS;
    #endif /* USE_TIMEOUT */

    while (LL_ADC_IsCalibrationOnGoing(ADC1) != 0)
    {
        #if (USE_TIMEOUT == 1)
        /* Check SysTick counter flag to decrement the time-out value */
        if (LL_SYSTICK_IsActiveCounterFlag())
        {
            }
        }
        #endif /* USE_TIMEOUT */
    }

    /* Delay between ADC end of calibration and ADC enable. */
    /* Note: Variable divided by 2 to compensate partially */
    /* CPU processing cycles (depends on compilation optimization). */
    wait_loop_index = (ADC_DELAY_CALIB_ENABLE_CPU_CYCLES >> 1);
    while(wait_loop_index != 0)
    {
        wait_loop_index--;
    }

    /* Enable ADC */
    LL_ADC_Enable(ADC1);

    /* Poll for ADC ready to convert */
    #if (USE_TIMEOUT == 1)
    Timeout = ADC_ENABLE_TIMEOUT_MS;
    #endif /* USE_TIMEOUT */
}
```

```

while (LL_ADC_IsActiveFlag_ADRDY(ADC1) == 0)
{
    #if (USE_TIMEOUT == 1)
        /* Check Systick counter flag to decrement the time-out value */
        if (LL_SYSTICK_IsActiveCounterFlag())
        {
        }
    #endif /* USE_TIMEOUT */
}

/* Note: ADC flag ADRDY is not cleared here to be able to check ADC      */
/*       status afterwards.                                              */
/*       This flag should be cleared at ADC Deactivation, before a new   */
/*       ADC activation, using function "LL_ADC_ClearFlag_ADRDY()".      */
}

/*## Operation on ADC hierarchical scope: ADC group regular #####*/
/* Note: No operation on ADC group regular performed here.             */
/*       ADC group regular conversions to be performed after this function */
/*       using function:                                                */
/*       "LL_ADC_REG_StartConversion();"                                */

/*## Operation on ADC hierarchical scope: ADC group injected #####*/
/* Note: No operation on ADC group injected performed here.             */
/*       ADC group injected conversions to be performed after this function */
/*       using function:                                                */
/*       "LL_ADC_INJ_StartConversion();"                                */

}

//Configure & Activate ADC
void setupADC(void)
{
    configureADC();
    activateADC();
}

/**
 * @brief ADC group regular overrun interruption callback
 * @note This function is executed when ADC group regular
 *       overrun error occurs.
 * @retval None
 */
void AdcGrpRegularOverrunError_Callback(void)
{
    /* Note: Disable ADC interruption that caused this error before entering in */
    /*       infinite loop below.                                              */

    /* Disable ADC group regular overrun interruption */
    LL_ADC_DisableIT_OVR(ADC1);
}

void AdcGrpRegularSequenceConvComplete_Callback(void)
{
    /* Update status variable of ADC group regular sequence */

```

```
ubAdcGrpRegularSequenceConvStatus = 1;  
ubAdcGrpRegularSequenceConvCount++;  
}
```


Dac_nano.h:

Librería escrita para el manejo del convertor digital anaógico (DAC), su inicialización y procesado.

```
#include "dac_nano.h"

/* NANO homebrew functions -----*/
void ConfigureDAC(void)
{
    /*## Configuration of GPIO used by DAC channels #####*/

    /* Enable GPIO Clock */
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOA);

    /* Configure GPIO in analog mode to be used as DAC output */
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_4, LL_GPIO_MODE_ANALOG);

    /*## Configuration of NVIC #####*/
    /* Configure NVIC to enable DAC1 interruptions */
    NVIC_SetPriority(TIM6_DAC1_IRQn, 0);
    NVIC_EnableIRQ(TIM6_DAC1_IRQn);

    /*## Configuration of DAC #####*/

    /* Enable DAC clock */
    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_DAC1);

    /* Select trigger source */
    LL_DAC_SetTriggerSource(DAC1, LL_DAC_CHANNEL_1, LL_DAC_TRIG_SOFTWARE);

    /* Set the output for the selected DAC channel */
    //LL_DAC_SetOutputBuffer(DAC1, LL_DAC_CHANNEL_1, LL_DAC_OUTPUT_BUFFER_ENABLE);

    /* Disable DAC channel DMA request */
    // LL_DAC_DisableDMAReq(DAC1, LL_DAC_CHANNEL_1);

    /* Set the data to be loaded in the data holding register */
    // LL_DAC_ConvertData12RightAligned(DAC1, LL_DAC_CHANNEL_1, 0x000);

    /* Enable interruption DAC channel1 underrun */
    LL_DAC_EnableIT_DMAUDR1(DAC1);
}

void ActivateDAC(void)
{
    __IO uint32_t wait_loop_index = 0;

    /* Enable DAC channel */
    LL_DAC_Enable(DAC1, LL_DAC_CHANNEL_1);

    /* Delay for DAC channel voltage settling time from DAC channel startup. */
    /* Compute number of CPU cycles to wait for, from delay in us. */
    /* Note: Variable divided by 2 to compensate partially */
    /* CPU processing cycles (depends on compilation optimization). */
}
```

```

/* Note: If system core clock frequency is below 200kHz, wait time */
/* is only a few CPU processing cycles. */
wait_loop_index = ((LL_DAC_DELAY_STARTUP_VOLTAGE_SETTLING_US * (SystemCoreClock / (100000
* 2))) / 10);
while(wait_loop_index != 0)
{
    wait_loop_index--;
}

/* Enable DAC channel trigger */
/* Note: DAC channel conversion can start from trigger enable: */
/* - if DAC channel trigger source is set to SW: */
/* DAC channel conversion will start after trig order */
/* using function "LL_DAC_TrigSWConversion()". */
/* - if DAC channel trigger source is set to external trigger */
/* (timer, ...): */
/* DAC channel conversion can start immediately */
/* (after next trig order from external trigger) */
LL_DAC_EnableTrigger(DAC1, LL_DAC_CHANNEL_1);

//Disable trigger to update DAC every data convert automatically
//LL_DAC_DisableTrigger(DAC1, LL_DAC_CHANNEL_1);

}

void setupDAC(void)
{
    ConfigureDAC();
    ActivateDAC();
}

void writeDAC(uint16_t data)
{
    /* Set the data to be loaded in the data holding register */
    LL_DAC_ConvertData12RightAligned(DAC1, LL_DAC_CHANNEL_1, data);

    /* Trig DAC conversion by software (Not necessary if trigger is disabled) */
    //LL_DAC_TrigSWConversion(DAC1, LL_DAC_CHANNEL_1);
}

```

Dma_nano.h:

Librería escrita para el manejo del acceso directo a memoria (DMA), su inicialización y procesado.

```
#include "dma_nano.h"

/**
 * @brief This function configures DMA for transfer of data from ADC
 * @param None
 * @retval None
 */
void configureDMA(uint32_t bufferName)
{
    /*## Configuration of NVIC #####*/
    /* Configure NVIC to enable DMA interruptions */
    NVIC_SetPriority(DMA1_Channel1_IRQn, 1); /* DMA IRQ lower priority than ADC IRQ */
    NVIC_EnableIRQ(DMA1_Channel1_IRQn);

    /*## Configuration of DMA #####*/
    /* Enable the peripheral clock of DMA */
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_DMA1);

    /* Configure the DMA transfer */
    /* - DMA transfer in circular mode to match with ADC configuration: */
    /*   DMA unlimited requests. */
    /* - DMA transfer from ADC without address increment. */
    /* - DMA transfer to memory with address increment. */
    /* - DMA transfer from ADC by half-word to match with ADC configuration: */
    /*   ADC resolution 12 bits. */
    /* - DMA transfer to memory by half-word to match with ADC conversion data */
    /*   buffer variable type: half-word. */
    LL_DMA_ConfigTransfer(DMA1,
        LL_DMA_CHANNEL_1,
        LL_DMA_DIRECTION_PERIPH_TO_MEMORY |
        LL_DMA_MODE_CIRCULAR |
        LL_DMA_PERIPH_NOINCREMENT |
        LL_DMA_MEMORY_INCREMENT |
        LL_DMA_PDATAALIGN_HALFWORD |
        LL_DMA_MDATAALIGN_HALFWORD |
        LL_DMA_PRIORITY_HIGH    );

    /* Set DMA transfer addresses of source and destination */
    LL_DMA_ConfigAddresses(DMA1,
        LL_DMA_CHANNEL_1,
        LL_ADC_DMA_GetRegAddr(ADC1, LL_ADC_DMA_REG_REGULAR_DATA),
        bufferName,
        LL_DMA_DIRECTION_PERIPH_TO_MEMORY);

    /* Set DMA transfer size */
    LL_DMA_SetDataLength(DMA1,
        LL_DMA_CHANNEL_1,
        ADC_CONVERTED_DATA_BUFFER_SIZE);

    /* Enable DMA transfer interruption: transfer complete */
    LL_DMA_EnableIT_TC(DMA1,
```

```
        LL_DMA_CHANNEL_1);

/* Enable DMA transfer interruption: transfer error */
LL_DMA_EnableIT_TE(DMA1,
                   LL_DMA_CHANNEL_1);

/*## Activation of DMA #####*/
/* Enable the DMA transfer */
LL_DMA_EnableChannel(DMA1,
                    LL_DMA_CHANNEL_1);
}
```

Exti_nano.h:

Librería escrita para el manejo de las interrupciones externas (EXTI), su inicialización y procesado.

```
#include "exti_nano.h"

/* main.c functions -----*/
void speedIRQHandler(void);
void waveIRQHandler(void);

/* NANO homebrew functions -----*/

/* Connect External Line to the GPIO */
void configureEXTI(void){

    /* Wave Select */
    LL_SYSCFG_SetEXTISource(LL_SYSCFG_EXTI_PORTA, LL_SYSCFG_EXTI_LINE10);

    /* Enable a rising trigger EXTI line 10 Interrupt */
    LL_EXTI_EnableIT_0_31(LL_EXTI_LINE_10);
    LL_EXTI_EnableRisingTrig_0_31(LL_EXTI_LINE_10);

    /* Configure NVIC for EXTI15_10_IRQn */
    NVIC_EnableIRQ(EXTI15_10_IRQn);
    NVIC_SetPriority(EXTI15_10_IRQn,0);
}

/* Handles external line 15_10 interrupt request */
void IRQHandler(void)
{
    /* Manage Wave Flags */
    if(LL_EXTI_IsActiveFlag_0_31(LL_EXTI_LINE_10) != RESET)
    {
        /* Manage code in main.c */
        waveIRQHandler();
        //Clean INT flag
        LL_EXTI_ClearFlag_0_31(LL_EXTI_LINE_10);
    }
}
```


Gpio_nano.h:

Librería escrita para el manejo de las entradas y salidas (GPIO), su inicialización y procesado.

```
#include "gpio_nano.h"

/* NANO homebrew functions -----*/

void configurePin(GPIO_TypeDef *GPIOPORT, uint32_t GPIOPIN, uint32_t GPIOMODE, uint32_t PULL)
{
    if (GPIOPORT == GPIOA){
        GPIOCLOCK = LL_AHB1_GRP1_PERIPH_GPIOA;}
    else if (GPIOPORT == GPIOB){
        GPIOCLOCK = LL_AHB1_GRP1_PERIPH_GPIOB;}
    else if (GPIOPORT == GPIOC){
        GPIOCLOCK = LL_AHB1_GRP1_PERIPH_GPIOC;}
    else if (GPIOPORT == GPIOD){
        GPIOCLOCK = LL_AHB1_GRP1_PERIPH_GPIOD;}

    /* Enable GPIO Clock */
    LL_AHB1_GRP1_EnableClock(GPIOCLOCK);

    /* Configure GPIO in analog mode to be used as ADC input */
    LL_GPIO_SetPinMode(GPIOPORT, GPIOPIN, GPIOMODE);

    if (GPIOMODE == 0){
        if (PULL){
            LL_GPIO_SetPinPull(GPIOPORT, GPIOPIN, LL_GPIO_PULL_UP);
        } else {
            LL_GPIO_SetPinPull(GPIOPORT, GPIOPIN, LL_GPIO_PULL_NO);}
    }
    else if (GPIOMODE == 1){
        LL_GPIO_SetPinOutputType(GPIOPORT, GPIOPIN, LL_GPIO_OUTPUT_PUSHPULL);
    }
}

// Set HIGH Pin
void setPin(GPIO_TypeDef *GPIOx, uint32_t PinMask)
{
    WRITE_REG(GPIOx->ODR, PinMask);
}

// Set LOW Pin
void resetPin(GPIO_TypeDef *GPIOx, uint32_t PinMask)
{
    WRITE_REG(GPIOx->ODR, !PinMask);
}

// Set HIGH/LOW Pin
void statePin(GPIO_TypeDef *GPIOx, uint32_t PinMask, uint32_t state)
{
    if (state){
        setPin(GPIOx, PinMask);
    }
}
```

```
        } else {
            resetPin(GPIOx, PinMask);
        }
    }

// Toggle Pin
void togglePin(GPIO_TypeDef *GPIOx, uint32_t PinMask)
{
    WRITE_REG(GPIOx->ODR, READ_REG(GPIOx->ODR) ^ PinMask);
}

// Read Pin
uint32_t readPin(GPIO_TypeDef *GPIOx, uint32_t PinMask)
{
    return (uint32_t)((READ_REG(GPIOx->IDR)) & PinMask);
}
```

Pwm_nano.h:

Librería escrita para el manejo del módulo PWM, su inicialización y procesado.

```
#include "pwm_nano.h"

void configurePWM(void)
{
    /******
    /* GPIO AF configuration */
    /******
    /* Enable the peripheral clock of GPIOs */
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOA);
        /* Enable the peripheral clock of GPIOs */
    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOB);

        /* GPIO TIM3_CH1 configuration */
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_6, LL_GPIO_MODE_ALTERNATE);
    LL_GPIO_SetPinPull(GPIOA, LL_GPIO_PIN_6, LL_GPIO_PULL_DOWN);
    LL_GPIO_SetPinSpeed(GPIOA, LL_GPIO_PIN_6, LL_GPIO_SPEED_FREQ_HIGH);
    LL_GPIO_SetAFPin_0_7(GPIOA, LL_GPIO_PIN_6, LL_GPIO_AF_2);

    /* GPIO TIM3_CH2 configuration */
    LL_GPIO_SetPinMode(GPIOA, LL_GPIO_PIN_7, LL_GPIO_MODE_ALTERNATE);
    LL_GPIO_SetPinPull(GPIOA, LL_GPIO_PIN_7, LL_GPIO_PULL_DOWN);
    LL_GPIO_SetPinSpeed(GPIOA, LL_GPIO_PIN_7, LL_GPIO_SPEED_FREQ_HIGH);
    LL_GPIO_SetAFPin_0_7(GPIOA, LL_GPIO_PIN_7, LL_GPIO_AF_2);

        /* GPIO TIM3_CH2 configuration */
    LL_GPIO_SetPinMode(GPIOB, LL_GPIO_PIN_0, LL_GPIO_MODE_ALTERNATE);
    LL_GPIO_SetPinPull(GPIOB, LL_GPIO_PIN_0, LL_GPIO_PULL_DOWN);
    LL_GPIO_SetPinSpeed(GPIOB, LL_GPIO_PIN_0, LL_GPIO_SPEED_FREQ_HIGH);
    LL_GPIO_SetAFPin_0_7(GPIOB, LL_GPIO_PIN_0, LL_GPIO_AF_2);

    /******
    /* Configure the NVIC to handle TIM3 interrupt */
    /******
    NVIC_SetPriority(TIM3_IRQn, 0);
    NVIC_EnableIRQ(TIM3_IRQn);

    /******
    /* Peripheral clocks enabling */
    /******
    /* Enable the timer peripheral clock */
    LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM3);

    /******
    /* Time base configuration */
    /******

    /* Set the pre-scaler value to have TIM3 counter clock equal to 72MHz */
    LL_TIM_SetPrescaler(TIM3, __LL_TIM_CALC_PSC(SystemCoreClock, 72000000));

    /* Enable TIM3_ARR register preload. Writing to or reading from the      */
    /* auto-reload register accesses the preload register. The content of the */
    /* preload register are transferred into the shadow register at each update */
}
```

```

/* event (UEV). */
LL_TIM_EnableARRPreload(TIM3);

/* Set the auto-reload value to have a counter frequency of 50 Hz */
/* TIM3CLK = SystemCoreClock / (APB prescaler & multiplier) */
LL_TIM_SetAutoReload(TIM3, __LL_TIM_CALC_ARR(SystemCoreClock, LL_TIM_GetPrescaler(TIM3),
50));

/*****/
/* Output waveform configuration */
/*****/
/* Set output mode */
/* Reset value is LL_TIM_OC_MODE_FROZEN */
LL_TIM_OC_SetMode(TIM3, LL_TIM_CHANNEL_CH1, LL_TIM_OC_MODE_PWM1);
LL_TIM_OC_SetMode(TIM3, LL_TIM_CHANNEL_CH2, LL_TIM_OC_MODE_PWM1);
LL_TIM_OC_SetMode(TIM3, LL_TIM_CHANNEL_CH3, LL_TIM_OC_MODE_PWM1);

/* Set compare value to half of the counter period (50% duty cycle) */
LL_TIM_OC_SetCompareCH1(TIM3, ( (LL_TIM_GetAutoReload(TIM3) + 1 ) / 2));
LL_TIM_OC_SetCompareCH2(TIM3, ( (LL_TIM_GetAutoReload(TIM3) + 1 ) / 2));
LL_TIM_OC_SetCompareCH3(TIM3, ( (LL_TIM_GetAutoReload(TIM3) + 1 ) / 2));

/* Enable TIM3 register preload. Read/Write operations access the */
/* preload register. TIM3 preload value is loaded in the active */
/* at each update event. */
LL_TIM_OC_EnablePreload(TIM3, LL_TIM_CHANNEL_CH1);
LL_TIM_OC_EnablePreload(TIM3, LL_TIM_CHANNEL_CH2);
LL_TIM_OC_EnablePreload(TIM3, LL_TIM_CHANNEL_CH3);

/*****/
/* TIM3 interrupts set-up */
/*****/
/* Enable the capture/compare interrupt */
LL_TIM_EnableIT_CC1(TIM3);
LL_TIM_EnableIT_CC2(TIM3);
LL_TIM_EnableIT_CC3(TIM3);

/*****/
/* Start output signal generation */
/*****/
/* Enable output channel 1 */
LL_TIM_CC_EnableChannel(TIM3, LL_TIM_CHANNEL_CH1);
/* Enable output channel 2 */
LL_TIM_CC_EnableChannel(TIM3, LL_TIM_CHANNEL_CH2);
/* Enable output channel 3 */
LL_TIM_CC_EnableChannel(TIM3, LL_TIM_CHANNEL_CH3);

/* Enable counter */
LL_TIM_EnableCounter(TIM3);

/* Force update generation */
LL_TIM_GenerateEvent_UPDATE(TIM3);
}

void configureDutyCycle(uint32_t channel ,uint32_t D)
{

```

Anexo Código

```
uint32_t P;    /* Pulse duration */
uint32_t T;    /* PWM signal period */

/* PWM signal period is determined by the value of the auto-reload register */
T = LL_TIM_GetAutoReload(TIM3) + 1;

/* Pulse duration is determined by the value of the compare register.      */
/* Its value is calculated in order to match the requested duty cycle.      */
P = ((D/41)*T)/100;

    if (channel == 0){
LL_TIM_OC_SetCompareCH1(TIM3, 0);
    LL_TIM_OC_SetCompareCH2(TIM3, 0);
    LL_TIM_OC_SetCompareCH3(TIM3, 0);}

    else if (channel == 1){
LL_TIM_OC_SetCompareCH1(TIM3, P);
    LL_TIM_OC_SetCompareCH2(TIM3, 0);
    LL_TIM_OC_SetCompareCH3(TIM3, 0);}

    else if (channel == 2){
LL_TIM_OC_SetCompareCH1(TIM3, 0);
    LL_TIM_OC_SetCompareCH2(TIM3, P);
    LL_TIM_OC_SetCompareCH3(TIM3, 0);}

    else if (channel == 3){
LL_TIM_OC_SetCompareCH1(TIM3, 0);
    LL_TIM_OC_SetCompareCH2(TIM3, 0);
    LL_TIM_OC_SetCompareCH3(TIM3, P);}

    else if (channel == 4){
LL_TIM_OC_SetCompareCH1(TIM3, P);
    LL_TIM_OC_SetCompareCH2(TIM3, P);
    LL_TIM_OC_SetCompareCH3(TIM3, 0);}

    else if (channel == 5){
LL_TIM_OC_SetCompareCH1(TIM3, P);
    LL_TIM_OC_SetCompareCH2(TIM3, 0);
    LL_TIM_OC_SetCompareCH3(TIM3, P);}

    else if (channel == 6){
LL_TIM_OC_SetCompareCH1(TIM3, 0);
    LL_TIM_OC_SetCompareCH2(TIM3, P);
    LL_TIM_OC_SetCompareCH3(TIM3, P);}

    else if (channel == 7){
LL_TIM_OC_SetCompareCH1(TIM3, P);
    LL_TIM_OC_SetCompareCH2(TIM3, P);
    LL_TIM_OC_SetCompareCH3(TIM3, P);}

    else if (channel == 8){
LL_TIM_OC_SetCompareCH1(TIM3, P);
    LL_TIM_OC_SetCompareCH2(TIM3, 0);
    LL_TIM_OC_SetCompareCH3(TIM3, P/8);}
```


Tim_nano.h:

Librería escrita para el manejo de los timers internos del microcontrolador (TIM), su inicialización y procesado.

```
#include "tim_nano.h"

/**
 * @brief System Clock Configuration
 * The system Clock is configured as follow :
 * System Clock source = PLL (HSI)
 * SYSCLK(Hz) = 72000000
 * HCLK(Hz) = 72000000
 * AHB Prescaler = 1
 * APB1 Prescaler = 2
 * APB2 Prescaler = 1
 * PLLMUL = 16
 * Flash Latency(WS) = 2
 * @param None
 * @retval None
 */
void configureClock(void)
{
    /* Set FLASH latency */
    LL_FLASH_SetLatency(LL_FLASH_LATENCY_2);

    /* Enable HSE if not already activated*/
    if (LL_RCC_HSE_IsReady() == 0)
    {
        /* Enable HSE and wait for activation*/
        LL_RCC_HSE_Enable();
        while(LL_RCC_HSE_IsReady() != 1)
        {
        };
    }

    /* Main PLL configuration and activation */
    LL_RCC_PLL_ConfigDomain_SYS(LL_RCC_PLLSOURCE_HSE_DIV_1, LL_RCC_PLL_MUL_9);

    LL_RCC_PLL_Enable();
    while(LL_RCC_PLL_IsReady() != 1)
    {
    };

    /* Sysclk activation on the main PLL */
    LL_RCC_SetAHBPrescaler(LL_RCC_SYSCLK_DIV_1);
    LL_RCC_SetSysClkSource(LL_RCC_SYS_CLKSOURCE_PLL);
    while(LL_RCC_GetSysClkSource() != LL_RCC_SYS_CLKSOURCE_STATUS_PLL)
    {
    };

    /* Set APB1 & APB2 prescaler*/
    LL_RCC_SetAPB1Prescaler(LL_RCC_APB1_DIV_2);
    LL_RCC_SetAPB2Prescaler(LL_RCC_APB2_DIV_1);
}
```

```

/* Set systick to 1ms in using frequency set to 72MHz */
/* This frequency can be calculated through LL RCC macro */
/* ex: __LL_RCC_CALC_PLLCLK_FREQ ((HSI_VALUE / 2), LL_RCC_PLL_MUL_16) */
LL_Init1msTick(72000000);

/* Update CMSIS variable (which can be updated also through SystemCoreClockUpdate
function) */
LL_SetSystemCoreClock(72000000);
}

//Configure & Activate TIM
void configureTIM(void){

/* Configure the NVIC to handle TIM2 update interrupt */
NVIC_SetPriority(TIM2_IRQn, 0);
NVIC_EnableIRQ(TIM2_IRQn);

/* Enable the timer peripheral clock */
LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_TIM2);

/* Set the pre-scaler value to have TIM2 counter clock equal to 72MHz */
LL_TIM_SetPrescaler(TIM2, __LL_TIM_CALC_PSC(SystemCoreClock, 72000000));

/* Enable TIM2_ARR register preload. Writing to or reading from the      */
/* auto-reload register accesses the preload register. The content of the */
/* preload register are transferred into the shadow register at each update */
/* event (UEV).                                                            */
LL_TIM_EnableARRPreload(TIM2);

/* Set the pre-scaler value to have TIM2 counter interrupt equal to 96kHz */
LL_TIM_SetAutoReload(TIM2, __LL_TIM_CALC_ARR(SystemCoreClock, LL_TIM_GetPrescaler(TIM3),
96000));

/* Enable the update interrupt */
LL_TIM_EnableIT_UPDATE(TIM2);

/* Enable counter */
LL_TIM_EnableCounter(TIM2);

/* Force update generation */
LL_TIM_GenerateEvent_UPDATE(TIM2);
}

```

2.1 Librerías externas

Algunas librerías que necesitamos ya están escritas por otras personas, por tanto vamos a incluirlas en nuestro proyecto ya que en los test han demostrado ser robustas y no han generado ningún problema. Es el caso de las siguientes librerías:

Wire.h:

Librería utilizada para el manejo de dispositivos i2c con Arduino. Cuenta con inicialización y procesado.

```

/*
  TwoWire.cpp - TWI/I2C library for Arduino & Wiring
  Copyright (c) 2006 Nicholas Zambetti. All right reserved.
  This library is free software; you can redistribute it and/or
  modify it under the terms of the GNU Lesser General Public
  License as published by the Free Software Foundation; either
  version 2.1 of the License, or (at your option) any later version.
  This library is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
  Lesser General Public License for more details.
  You should have received a copy of the GNU Lesser General Public
  License along with this library; if not, write to the Free Software
  Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
  Modified 2012 by Todd Krein (todd@krein.org) to implement repeated starts
  Modified December 2014 by Ivan Grokhotkov (ivan@esp8266.com) - esp8266 support
  Modified April 2015 by Hrsto Gochkov (ficeto@ficeto.com) - alternative esp8266 support
*/

extern "C" {
  #include <stdlib.h>
  #include <string.h>
  #include <inttypes.h>
}

#include "twi.h"
#include "Wire.h"

//Some boards don't have these pins available, and hence don't support Wire.
//Check here for compile-time error.
#if !defined(PIN_WIRE_SDA) || !defined(PIN_WIRE_SCL)
#error Wire library is not supported on this board
#endif

// Initialize Class Variables //////////////////////////////////////

uint8_t TwoWire::rxBuffer[BUFFER_LENGTH];
uint8_t TwoWire::rxBufferIndex = 0;
uint8_t TwoWire::rxBufferLength = 0;

```

```

uint8_t TwoWire::txAddress = 0;
uint8_t TwoWire::txBuffer[BUFFER_LENGTH];
uint8_t TwoWire::txBufferIndex = 0;
uint8_t TwoWire::txBufferLength = 0;

uint8_t TwoWire::transmitting = 0;
void (*TwoWire::user_onRequest)(void);
void (*TwoWire::user_onReceive)(int);

static int default_sda_pin = SDA;
static int default_scl_pin = SCL;

// Constructors ////////////////////////////////////////////////////////////////////

TwoWire::TwoWire(){}

// Public Methods ////////////////////////////////////////////////////////////////////

void TwoWire::begin(int sda, int scl){
    default_sda_pin = sda;
    default_scl_pin = scl;
    twi_init(sda, scl);
    flush();
}

void TwoWire::pins(int sda, int scl){
    default_sda_pin = sda;
    default_scl_pin = scl;
}

void TwoWire::begin(void){
    begin(default_sda_pin, default_scl_pin);
}

void TwoWire::begin(uint8_t address){
    (void)address;
    // twi_setAddress(address);
    // twi_attachSlaveTxEvent(onRequestService);
    // twi_attachSlaveRxEvent(onReceiveService);
    begin();
}

uint8_t TwoWire::status(){
    return twi_status();
}

void TwoWire::begin(int address){
    begin((uint8_t)address);
}

void TwoWire::setClock(uint32_t frequency){
    twi_setClock(frequency);
}

void TwoWire::setClockStretchLimit(uint32_t limit){
    twi_setClockStretchLimit(limit);
}

```

```

size_t TwoWire::requestFrom(uint8_t address, size_t size, bool sendStop){
    if(size > BUFFER_LENGTH){
        size = BUFFER_LENGTH;
    }
    size_t read = (twi_readFrom(address, rxBuffer, size, sendStop) == 0)?size:0;
    rxBufferIndex = 0;
    rxBufferLength = read;
    return read;
}

uint8_t TwoWire::requestFrom(uint8_t address, uint8_t quantity, uint8_t sendStop){
    return requestFrom(address, static_cast<size_t>(quantity), static_cast<bool>(sendStop));
}

uint8_t TwoWire::requestFrom(uint8_t address, uint8_t quantity){
    return requestFrom(address, static_cast<size_t>(quantity), true);
}

uint8_t TwoWire::requestFrom(int address, int quantity){
    return requestFrom(static_cast<uint8_t>(address), static_cast<size_t>(quantity), true);
}

uint8_t TwoWire::requestFrom(int address, int quantity, int sendStop){
    return requestFrom(static_cast<uint8_t>(address), static_cast<size_t>(quantity),
static_cast<bool>(sendStop));
}

void TwoWire::beginTransaction(uint8_t address){
    transmitting = 1;
    txAddress = address;
    txBufferIndex = 0;
    txBufferLength = 0;
}

void TwoWire::beginTransaction(int address){
    beginTransmission((uint8_t)address);
}

uint8_t TwoWire::endTransmission(uint8_t sendStop){
    int8_t ret = twi_writeTo(txAddress, txBuffer, txBufferLength, sendStop);
    txBufferIndex = 0;
    txBufferLength = 0;
    transmitting = 0;
    return ret;
}

uint8_t TwoWire::endTransmission(void){
    return endTransmission(true);
}

size_t TwoWire::write(uint8_t data){
    if(transmitting){
        if(txBufferLength >= BUFFER_LENGTH){
            setWriteError();
            return 0;
        }
    }
}

```

```

    txBuffer[txBufferIndex] = data;
    ++txBufferIndex;
    txBufferLength = txBufferIndex;
} else {
    // i2c_slave_transmit(&data, 1);
}
return 1;
}

size_t TwoWire::write(const uint8_t *data, size_t quantity){
    if(transmitting){
        for(size_t i = 0; i < quantity; ++i){
            if(!write(data[i])) return i;
        }
    }else{
        // i2c_slave_transmit(data, quantity);
    }
    return quantity;
}

int TwoWire::available(void){
    int result = rxBufferLength - rxBufferIndex;

    if (!result) {
        // yielding here will not make more data "available",
        // but it will prevent the system from going into WDT reset
        optimistic_yield(1000);
    }

    return result;
}

int TwoWire::read(void){
    int value = -1;
    if(rxBufferIndex < rxBufferLength){
        value = rxBuffer[rxBufferIndex];
        ++rxBufferIndex;
    }
    return value;
}

int TwoWire::peek(void){
    int value = -1;
    if(rxBufferIndex < rxBufferLength){
        value = rxBuffer[rxBufferIndex];
    }
    return value;
}

void TwoWire::flush(void){
    rxBufferIndex = 0;
    rxBufferLength = 0;
    txBufferIndex = 0;
    txBufferLength = 0;
}

void TwoWire::onReceiveService(uint8_t* inBytes, int numBytes)

```

```

{
  (void)inBytes;
  (void)numBytes;
  // don't bother if user hasn't registered a callback
  // if(!user_onReceive){
  //   return;
  // }
  // // don't bother if rx buffer is in use by a master requestFrom() op
  // // i know this drops data, but it allows for slight stupidity
  // // meaning, they may not have read all the master requestFrom() data yet
  // if(rxBufferIndex < rxBufferLength){
  //   return;
  // }
  // // copy twi rx buffer into local read buffer
  // // this enables new reads to happen in parallel
  // for(uint8_t i = 0; i < numBytes; ++i){
  //   rxBuffer[i] = inBytes[i];
  // }
  // // set rx iterator vars
  // rxBufferIndex = 0;
  // rxBufferLength = numBytes;
  // // alert user program
  // user_onReceive(numBytes);
}

void TwoWire::onRequestService(void){
  // // don't bother if user hasn't registered a callback
  // if(!user_onRequest){
  //   return;
  // }
  // // reset tx buffer iterator vars
  // // !!! this will kill any pending pre-master sendTo() activity
  // txBufferIndex = 0;
  // txBufferLength = 0;
  // // alert user program
  // user_onRequest();
}

void TwoWire::onReceive( void (*function)(int) ){
  (void)function;
  //user_onReceive = function;
}

void TwoWire::onRequest( void (*function)(void) ){
  (void)function;
  //user_onRequest = function;
}

// Preinstantiate Objects //////////////////////////////////////

#ifdef NO_GLOBAL_INSTANCES && !defined(NO_GLOBAL_TWOWIRE)
TwoWire Wire;
#endif

```


Adafruit_MCP4725.h:

Librería escrita para el manejo del DAC por i2c MCP4725, incluyendo su inicialización y su procesado para realizar una conversión digital-analógica.

```

/*****
/*!
  @file    Adafruit_MCP4725.cpp
  @author  K.Townsend (Adafruit Industries)
  @license BSD (see license.txt)

  I2C Driver for Microchip's MCP4725 I2C DAC
  This is a library for the Adafruit MCP4725 breakout
  ----> http://www.adafruit.com/products/935

  Adafruit invests time and resources providing this open source code,
  please support Adafruit and open-source hardware by purchasing
  products from Adafruit!
  @section HISTORY
  v1.0 - First release
*/
*****/
#if ARDUINO >= 100
  #include "Arduino.h"
#else
  #include "WProgram.h"
#endif

#include <Wire.h>

#include "Adafruit_MCP4725.h"

/*****
/*!
  @brief  Instantiates a new MCP4725 class
*/
*****/
Adafruit_MCP4725::Adafruit_MCP4725() {
}

/*****
/*!
  @brief  Setups the HW
*/
*****/
void Adafruit_MCP4725::begin(uint8_t addr) {
  _i2caddr = addr;
  Wire.begin();
}

/*****
/*!
  @brief  Sets the output voltage to a fraction of source vref. (Value
         can be 0..4095)
  @param[in] output

```

```

        The 12-bit value representing the relationship between
        the DAC's input voltage and its output voltage.
    @param[in] writeEEPROM
        If this value is true, 'output' will also be written
        to the MCP4725's internal non-volatile memory, meaning
        that the DAC will retain the current voltage output
        after power-down or reset.
*/
/*****
void Adafruit_MCP4725::setVoltage( uint16_t output, bool writeEEPROM )
{
#ifdef TWBR
    uint8_t twbrback = TWBR;
    TWBR = ((F_CPU / 400000L) - 16) / 2; // Set I2C frequency to 400kHz
#endif
    Wire.beginTransmission(_i2caddr);
    if (writeEEPROM)
    {
        Wire.write(MCP4726_CMD_WRITEDACEEPROM);
    }
    else
    {
        Wire.write(MCP4726_CMD_WRITEDAC);
    }
    Wire.write(output / 16);           // Upper data bits
    (D11.D10.D9.D8.D7.D6.D5.D4)
    Wire.write((output % 16) << 4);   // Lower data bits
    (D3.D2.D1.D0.x.x.x.x)
    Wire.endTransmission();
#ifdef TWBR
    TWBR = twbrback;
#endif
}

```

3. Código de los módulos

De los 5 módulos que se compone este proyecto, solo 3 módulos son de naturaleza digital, siendo el **VCO**, **LFO** y **Envolvente**. Esto quiere decir el código presente en este apartado es el correspondiente a estos módulos.

3.1. VCO

```

/* Includes -----*/
#include "main.h"

#define BRIGNTHNESS 1000

#define P_WIDTH      0

#define E_NUMBER      2.718281828459045235360

const uint16_t sineTable[256] =
{2048,2098,2148,2198,2248,2298,2348,2398,2447,2496,2545,2594,2642,2690,2737,2784,
2831,2877,2923,2968,3013,3057,3100,3143,3185,3226,3267,3307,3346,3385,3423,3459,
3495,3530,3565,3598,3630,3662,3692,3722,3750,3777,3804,3829,3853,3876,3898,3919,
3939,3958,3975,3992,4007,4021,4034,4045,4056,4065,4073,4080,4085,4089,4093,4094,
4095,4094,4093,4089,4085,4080,4073,4065,4056,4045,4034,4021,4007,3992,3975,3958,
3939,3919,3898,3876,3853,3829,3804,3777,3750,3722,3692,3662,3630,3598,3565,3530,
3495,3459,3423,3385,3346,3307,3267,3226,3185,3143,3100,3057,3013,2968,2923,2877,
2831,2784,2737,2690,2642,2594,2545,2496,2447,2398,2348,2298,2248,2198,2148,2098,
2048,1997,1947,1897,1847,1797,1747,1697,1648,1599,1550,1501,1453,1405,1358,1311,
1264,1218,1172,1127,1082,1038,995,952,910,869,828,788,749,710,672,636,
600,565,530,497,465,433,403,373,345,318,291,266,242,219,197,176,
156,137,120,103,88,74,61,50,39,30,22,15,10,6,2,1,
0,1,2,6,10,15,22,30,39,50,61,74,88,103,120,137,
156,176,197,219,242,266,291,318,345,373,403,433,465,497,530,565,
600,636,672,710,749,788,828,869,910,952,995,1038,1082,1127,1172,1218,
1264,1311,1358,1405,1453,1501,1550,1599,1648,1697,1747,1797,1847,1897,1947,1997};

const uint16_t rectisineTable[256] =
{2048,2073,2098,2123,2148,2173,2198,2223,2248,2273,2298,2323,2348,2373,2398,2422,
2447,2472,2496,2521,2545,2569,2594,2618,2642,2666,2690,2714,2737,2761,2784,2808,
2831,2854,2877,2900,2923,2946,2968,2990,3013,3035,3057,3078,3100,3122,3143,3164,
3185,3206,3226,3247,3267,3287,3307,3327,3346,3366,3385,3404,3423,3441,3459,3477,
3495,3513,3530,3548,3565,3581,3598,3614,3630,3646,3662,3677,3692,3707,3722,3736,
3750,3764,3777,3791,3804,3816,3829,3841,3853,3865,3876,3888,3898,3909,3919,3929,
3939,3949,3958,3967,3975,3984,3992,3999,4007,4014,4021,4027,4034,4040,4045,4051,
4056,4060,4065,4069,4073,4076,4080,4083,4085,4087,4089,4091,4093,4094,4094,4095,
4095,4095,4094,4094,4093,4091,4089,4087,4085,4083,4080,4076,4073,4069,4065,4060,
4056,4051,4045,4040,4034,4027,4021,4014,4007,3999,3992,3984,3975,3967,3958,3949,
3939,3929,3919,3909,3898,3888,3876,3865,3853,3841,3829,3816,3804,3791,3777,3764,
3750,3736,3722,3707,3692,3677,3662,3646,3630,3614,3598,3581,3565,3548,3530,3513,
3495,3477,3459,3441,3423,3404,3385,3366,3346,3327,3307,3287,3267,3247,3226,3206,
3185,3164,3143,3122,3100,3078,3057,3035,3013,2990,2968,2946,2923,2900,2877,2854,
2831,2808,2784,2761,2737,2714,2690,2666,2642,2618,2594,2569,2545,2521,2496,2472,
2447,2422,2398,2373,2348,2323,2298,2273,2248,2223,2198,2173,2148,2123,2098,2073};

```

```

/* Private function prototypes -----*/
void    Configure_ADC(void);
void    Activate_ADC(void);
void    ConversionStartPoll_ADC_GrpRegular(void);
void    LED_Init(void);
void    LED_On(void);
void    LED_Off(void);
void    LED_Blinking(uint32_t Period);
void    UserButton_Init(void);
void    Configure_TIMTimeBase(void);

/* Extern functions -----*/
/* GPIO NANO homebrew function prototypes -----*/
extern void    configurePin(GPIO_TypeDef *GPIOx, uint32_t PIN, uint32_t
GPIO_MODE, uint32_t PULL);
extern void    setPin(GPIO_TypeDef *GPIOx, uint32_t PinMask);
extern void    resetPin(GPIO_TypeDef *GPIOx, uint32_t PinMask);
extern void    statePin(GPIO_TypeDef *GPIOx, uint32_t PinMask, uint32_t
state);
extern void    togglePin(GPIO_TypeDef *GPIOx, uint32_t PinMask);
extern uint32_t readPin(GPIO_TypeDef *GPIOx, uint32_t PinMask);
/* ADC NANO homebrew function prototypes -----*/
extern void    configureADC(void);
extern void    activateADC(void);
extern void    setupADC(void);
extern void    convertADC(void);
/* DAC NANO homebrew function prototypes -----*/
extern void    configureDAC(void);
extern void    activateDAC(void);
extern void    setupDAC(void);
extern void    writeDAC(uint16_t value);
/* DMA NANO homebrew function prototypes -----*/
extern void    configureDMA(uint32_t bufferName);
/* TIM NANO homebrew function prototypes -----*/
extern void    configureTIM(void);
extern void    configureClock(void);
/* VCO NANO homebrew function prototypes -----*/
extern uint16_t voltsToFrequency(uint16_t voltsOctave);
extern uint32_t getDelay(uint32_t voltsOctave);
extern uint32_t map(float in, float inMin, float inMax, float outMin, float outMax);
extern uint32_t mapLog(float in, float inMin, float inMax, float outMin, float outMax);
extern uint16_t triangleWave(uint16_t waveState, uint16_t DACValue);
extern uint16_t sawWave(uint16_t waveState, uint16_t DACValue);
extern uint16_t squareWave(uint16_t waveState, uint16_t DACValue, uint16_t pWidth);
extern uint16_t pulseWave(uint16_t waveState, uint16_t DACValue);
/* PWM NANO homebrew function prototypes -----*/
extern void    configurePWM(void);
extern void    configureDutyCycle(uint32_t channel, uint32_t D);
/* EXTI NANO homebrew function prototypes -----*/
extern void    configureEXTI(void);
extern void    IRQHandler(void);

/* MAIN NANO homebrew function prototypes -----*/
void frequencyCalculations(void);
void testWaveform(uint16_t steps);
void testWavetable(void);

```

Anexo Código

```
void tickDelay(uint32_t ticks);
void countDelay(uint32_t counts);
uint32_t map(float in, float inMin, float inMax, float outMin, float outMax);

#define CLOCK 72000000

#define VDDA_APPLI ((uint32_t)3300)
#define ADC_CONVERTED_DATA_BUFFER_SIZE ((uint32_t) 3)

/* Variable to report status of DMA transfer of ADC group regular conversions */
/* 0: DMA transfer is not completed */
/* 1: DMA transfer is completed */
/* 2: DMA transfer has not been started yet (initial state) */
uint8_t ubDmaTransferStatus = 2; /* Variable set into DMA interruption callback */

/* VCO NANO Variables -----*/
volatile uint16_t voltsOctave, frecuencyWave, frecuencyTest, sampleRate, wave;
//ADC group regular conversion data
volatile uint16_t convertedADC[ADC_CONVERTED_DATA_BUFFER_SIZE];
//ADC Readings
volatile uint16_t A0,A1,A2, waveLength;

volatile uint32_t ticks = 0;
volatile uint32_t tick = 0;
volatile uint32_t tickEnd = 0;
volatile uint32_t delay = 50;
volatile uint32_t testLog = 0;
volatile uint32_t loopCounter = 0;
volatile uint32_t delayCounter = 0;
volatile uint32_t wavePin = 0, waveState = 0, waveCycles = 0;
volatile uint32_t DACValue = 0;
volatile uint32_t pWidth = 50;
volatile uint32_t timerTicks = 0;
volatile uint32_t timerARR = 8500;

volatile double exponent = 0;

/**
 * @brief Main program
 * @param None
 * @retval None
 */
int main(void)
{
    /* Configure the system clock to 72 MHz */
    configureClock();

    /* Configure DAC Output timer to 96kHz */
    configureTIM();

    /* Configure EXTI to select waveform */
    configureEXTI();

    /* Initialize RGB LED */
    configurePWM();
}
```

```

    /* Initialize pin to select the waveform by external trigger or button */
    configurePin(GPIOA, PIN10, INPUT, NOPULL);

    /* Configure DMA for data transfer from ADC */
    configureDMA((uint32_t)&convertedADC);

    /* Initialize A0 ADC - Waveform selector / A1 ADC - 1V/Oct input */
    setupADC();

    /* Initialize A2 DAC with trigger conversion */
    setupDAC();

    /* Infinite loop */
    while (1)
    {
        //Update ADC values
        convertADC();

    }
}

void countDelay(uint32_t delayCounter){
    uint32_t counter;
    for (counter = 0 ; counter < delayCounter ; counter++) {
    }
}

void convertADC(void){

    LL_ADC_REG_StartConversion(ADC1);

    /* Computation of ADC conversions */
    A0 = __LL_ADC_CALC_DATA_TO_VOLTAGE(3300, convertedADC[0], LL_ADC_RESOLUTION_12B);
    A1 = convertedADC[1];
    A2 = convertedADC[2];
}

void calculateARR(void){

    //Update TIM2 INT frequency
    if (P_WIDTH){
        exponent = 8597*pow(E_NUMBER, -0.00105 * A0)- A1*2;
        pWidth = map(A2, 0, 4095, 1, 512);
    }
    else{
        exponent = 8597*pow(E_NUMBER, -0.00105 * A0)- A1*2 - 1.5*A2;
    }

    //Prevent INT Blocking
    if (exponent < 175) { exponent = 175; }

    //Update TIM2 INT
    LL_TIM_SetAutoReload(TIM2, exponent);
}

void AdcDmaTransferComplete_Callback()
{

```

```

/* Update status variable of DMA transfer */
ubDmaTransferStatus = 1;
}

/**
 * @brief This function handles DMA1 interrupt request.
 * @param None
 * @retval None
 */
void DMA1_Channel1_IRQHandler(void)
{
    /* Check whether DMA transfer complete caused the DMA interruption */
    if(LL_DMA_IsActiveFlag_TC1(DMA1) == 1)
    {
        /* Clear flag DMA global interrupt */
        /* (global interrupt flag: half transfer and transfer complete flags) */
        LL_DMA_ClearFlag_GI1(DMA1);

        /* Call interruption treatment function */
        AdcDmaTransferComplete_Callback();
    }
}

void TIM3_IRQHandler(void)
{
    /* Check whether CC1 interrupt is pending */
    if(LL_TIM_IsActiveFlag_CC1(TIM3) == 1)
    {
        /* Clear the update interrupt flag*/
        LL_TIM_ClearFlag_CC1(TIM3);

        if (wave == 0) {configureDutyCycle(1, BRIGTHNESS);}
        else if(wave == 1) {configureDutyCycle(2, BRIGTHNESS);}
        else if (wave == 2) {configureDutyCycle(3, BRIGTHNESS);}
        else if (wave == 3) {configureDutyCycle(4, BRIGTHNESS);}
        else if (wave == 4) {configureDutyCycle(5, BRIGTHNESS);}
        else if (wave == 5) {configureDutyCycle(6, BRIGTHNESS);}
        else if (wave == 6) {configureDutyCycle(7, BRIGTHNESS);}
        else if (wave == 7) {configureDutyCycle(8, BRIGTHNESS);}
        else if (wave == 8) {configureDutyCycle(9, BRIGTHNESS);}
        else if (wave == 9) {configureDutyCycle(10, BRIGTHNESS);}
    }
    /* Check whether CC2 interrupt is pending */
    if(LL_TIM_IsActiveFlag_CC2(TIM3) == 1)
    {
        /* Clear the update interrupt flag*/
        LL_TIM_ClearFlag_CC2(TIM3);
    }
    /* Check whether CC3 interrupt is pending */
    if(LL_TIM_IsActiveFlag_CC3(TIM3) == 1)
    {
        /* Clear the update interrupt flag*/
        LL_TIM_ClearFlag_CC3(TIM3);
    }
}
}

```

```

/**
 * @brief TIM2 update interrupt processing
 * @param None
 * @retval None
 */
void TimerUpdate_Callback(void)
{
    if (wave == 0){
        //SINE WAVE
        DACValue = sineTable[waveState];
    }

    else if (wave == 1) {
        //RECTIFIED SINE WAVE
        DACValue = rectisineTable[waveState];
    }

    else if (wave == 2) {
        //SUM WAVE
        DACValue = (sineTable[waveState])-rectisineTable[waveState]/2;
    }

    else if (wave == 3) {
        //TRIANGLE WAVE
        DACValue = triangleWave(waveState,DACValue);
    }

    else if (wave == 4) {
        //SAW WAVE
        DACValue = sawWave(waveState,DACValue);
    }

    else if (wave == 5) {
        //SQUARE WAVE
        DACValue = squareWave(waveState,DACValue, pWidth);
    }

    else if (wave == 6) {
        //PULSE WAVE
        DACValue = pulseWave(waveState,DACValue);
    }

    //Set DAC output value;
    LL_DAC_ConvertData12RightAligned(DAC1, LL_DAC_CHANNEL_1, DACValue);

    //Update DAC Output
    LL_DAC_TrigSWConversion(DAC1, LL_DAC_CHANNEL_1);

    if (waveState < 255){ waveState++; }
    else {
        waveState = 0;
        waveCycles++;

        if(waveCycles >= 1024){
            //Calculate TIM2 Autorreload
            calculateARR();
        }
    }
}

```

```

    }
}

void waveIRQHandler(void){
    //Button debounce
    countDelay(1000000);
    //Wave polling
    if (wave <= 5){ wave++; } else { wave = 0; }
    //Clean phase counter
    waveState = 0;
    //Clean DAC value
    DACValue = 0;
}

#ifdef USE_FULL_ASSERT

/**
 * @brief Reports the name of the source file and the source line number
 *        where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(char *file, uint32_t line)
{
    /* User can add his own implementation to report the file name and line number,
       ex: printf("Wrong parameters value: file %s on line %d", file, line) */

    /* Infinite loop */
    while (1)
    {
    }
}
#endif

```


3.2. LFO

```

/* Includes -----*/
#include "main.h"

#define INT_RESET          0

uint16_t sineTable[4096] =
{2048,2051,2054,2057,2060,2063,2066,2069,2073,2076,2079,2082,2085,2088,2091,2095,
2098,2101,2104,2107,2110,2113,2117,2120,2123,2126,2129,2132,2135,2139,2142,2145,
2148,2151,2154,2157,2161,2164,2167,2170,2173,2176,2179,2182,2186,2189,2192,2195,
2198,2201,2204,2208,2211,2214,2217,2220,2223,2226,2229,2233,2236,2239,2242,2245,
2248,2251,2254,2258,2261,2264,2267,2270,2273,2276,2279,2283,2286,2289,2292,2295,
2298,2301,2304,2308,2311,2314,2317,2320,2323,2326,2329,2332,2336,2339,2342,2345,
2348,2351,2354,2357,2360,2364,2367,2370,2373,2376,2379,2382,2385,2388,2391,2395,
2398,2401,2404,2407,2410,2413,2416,2419,2422,2425,2429,2432,2435,2438,2441,2444,
2447,2450,2453,2456,2459,2462,2466,2469,2472,2475,2478,2481,2484,2487,2490,2493,
2496,2499,2502,2505,2508,2512,2515,2518,2521,2524,2527,2530,2533,2536,2539,2542,
2545,2548,2551,2554,2557,2560,2563,2566,2569,2572,2576,2579,2582,2585,2588,2591,
2594,2597,2600,2603,2606,2609,2612,2615,2618,2621,2624,2627,2630,2633,2636,2639,
2642,2645,2648,2651,2654,2657,2660,2663,2666,2669,2672,2675,2678,2681,2684,2687,
2690,2693,2696,2699,2702,2705,2708,2711,2714,2717,2720,2723,2726,2729,2732,2734,
2737,2740,2743,2746,2749,2752,2755,2758,2761,2764,2767,2770,2773,2776,2779,2782,
2785,2787,2790,2793,2796,2799,2802,2805,2808,2811,2814,2817,2820,2823,2825,2828,
2831,2834,2837,2840,2843,2846,2849,2851,2854,2857,2860,2863,2866,2869,2872,2875,
2877,2880,2883,2886,2889,2892,2895,2897,2900,2903,2906,2909,2912,2915,2917,2920,
2923,2926,2929,2932,2934,2937,2940,2943,2946,2949,2951,2954,2957,2960,2963,2965,
2968,2971,2974,2977,2979,2982,2985,2988,2991,2993,2996,2999,3002,3005,3007,3010,
3013,3016,3018,3021,3024,3027,3029,3032,3035,3038,3040,3043,3046,3049,3051,3054,
3057,3060,3062,3065,3068,3071,3073,3076,3079,3081,3084,3087,3090,3092,3095,3098,
3100,3103,3106,3108,3111,3114,3116,3119,3122,3125,3127,3130,3133,3135,3138,3140,
3143,3146,3148,3151,3154,3156,3159,3162,3164,3167,3170,3172,3175,3177,3180,3183,
3185,3188,3190,3193,3196,3198,3201,3203,3206,3209,3211,3214,3216,3219,3222,3224,
3227,3229,3232,3234,3237,3240,3242,3245,3247,3250,3252,3255,3257,3260,3262,3265,
3267,3270,3272,3275,3278,3280,3283,3285,3288,3290,3293,3295,3298,3300,3302,3305,
3307,3310,3312,3315,3317,3320,3322,3325,3327,3330,3332,3335,3337,3339,3342,3344,
3347,3349,3352,3354,3356,3359,3361,3364,3366,3368,3371,3373,3376,3378,3380,3383,
3385,3388,3390,3392,3395,3397,3399,3402,3404,3406,3409,3411,3413,3416,3418,3420,
3423,3425,3427,3430,3432,3434,3437,3439,3441,3444,3446,3448,3450,3453,3455,3457,
3460,3462,3464,3466,3469,3471,3473,3475,3478,3480,3482,3484,3487,3489,3491,3493,
3496,3498,3500,3502,3504,3507,3509,3511,3513,3515,3518,3520,3522,3524,3526,3529,
3531,3533,3535,3537,3539,3541,3544,3546,3548,3550,3552,3554,3556,3559,3561,3563,
3565,3567,3569,3571,3573,3575,3577,3580,3582,3584,3586,3588,3590,3592,3594,3596,
3598,3600,3602,3604,3606,3608,3610,3612,3614,3616,3618,3621,3623,3625,3627,3629,
3631,3633,3634,3636,3638,3640,3642,3644,3646,3648,3650,3652,3654,3656,3658,3660,
3662,3664,3666,3668,3670,3672,3673,3675,3677,3679,3681,3683,3685,3687,3689,3690,
3692,3694,3696,3698,3700,3702,3704,3705,3707,3709,3711,3713,3715,3716,3718,3720,
3722,3724,3725,3727,3729,3731,3733,3734,3736,3738,3740,3741,3743,3745,3747,3748,
3750,3752,3754,3755,3757,3759,3761,3762,3764,3766,3767,3769,3771,3773,3774,3776,
3778,3779,3781,3783,3784,3786,3788,3789,3791,3793,3794,3796,3797,3799,3801,3802,
3804,3806,3807,3809,3810,3812,3814,3815,3817,3818,3820,3821,3823,3825,3826,3828,
3829,3831,3832,3834,3835,3837,3838,3840,3842,3843,3845,3846,3848,3849,3851,3852,
3853,3855,3856,3858,3859,3861,3862,3864,3865,3867,3868,3870,3871,3872,3874,3875,
3877,3878,3879,3881,3882,3884,3885,3886,3888,3889,3891,3892,3893,3895,3896,3897,
3899,3900,3901,3903,3904,3905,3907,3908,3909,3911,3912,3913,3914,3916,3917,3918,
3920,3921,3922,3923,3925,3926,3927,3928,3930,3931,3932,3933,3935,3936,3937,3938,
3939,3941,3942,3943,3944,3945,3946,3948,3949,3950,3951,3952,3953,3955,3956,3957,

```


Anexo Código

3099, 3096, 3094, 3091, 3088, 3086, 3083, 3080, 3077, 3075, 3072, 3069, 3066, 3064, 3061, 3058, 3056, 3053, 3050, 3047, 3045, 3042, 3039, 3036, 3034, 3031, 3028, 3025, 3023, 3020, 3017, 3014, 3012, 3009, 3006, 3003, 3000, 2998, 2995, 2992, 2989, 2986, 2984, 2981, 2978, 2975, 2972, 2970, 2967, 2964, 2961, 2958, 2956, 2953, 2950, 2947, 2944, 2942, 2939, 2936, 2933, 2930, 2927, 2925, 2922, 2919, 2916, 2913, 2910, 2907, 2905, 2902, 2899, 2896, 2893, 2890, 2887, 2885, 2882, 2879, 2876, 2873, 2870, 2867, 2864, 2862, 2859, 2856, 2853, 2850, 2847, 2844, 2841, 2838, 2836, 2833, 2830, 2827, 2824, 2821, 2818, 2815, 2812, 2809, 2806, 2804, 2801, 2798, 2795, 2792, 2789, 2786, 2783, 2780, 2777, 2774, 2771, 2768, 2765, 2763, 2760, 2757, 2754, 2751, 2748, 2745, 2742, 2739, 2736, 2733, 2730, 2727, 2724, 2721, 2718, 2715, 2712, 2709, 2706, 2703, 2700, 2697, 2694, 2691, 2688, 2685, 2682, 2679, 2676, 2673, 2670, 2668, 2665, 2662, 2659, 2656, 2653, 2650, 2647, 2644, 2640, 2637, 2634, 2631, 2628, 2625, 2622, 2619, 2616, 2613, 2610, 2607, 2604, 2601, 2598, 2595, 2592, 2589, 2586, 2583, 2580, 2577, 2574, 2571, 2568, 2565, 2562, 2559, 2556, 2553, 2550, 2547, 2544, 2541, 2537, 2534, 2531, 2528, 2525, 2522, 2519, 2516, 2513, 2510, 2507, 2504, 2501, 2498, 2495, 2492, 2489, 2485, 2482, 2479, 2476, 2473, 2470, 2467, 2464, 2461, 2458, 2455, 2452, 2449, 2446, 2442, 2439, 2436, 2433, 2430, 2427, 2424, 2421, 2418, 2415, 2412, 2408, 2405, 2402, 2399, 2396, 2393, 2390, 2387, 2384, 2381, 2377, 2374, 2371, 2368, 2365, 2362, 2359, 2356, 2353, 2350, 2346, 2343, 2340, 2337, 2334, 2331, 2328, 2325, 2322, 2318, 2315, 2312, 2309, 2306, 2303, 2300, 2297, 2294, 2290, 2287, 2284, 2281, 2278, 2275, 2272, 2269, 2265, 2262, 2259, 2256, 2253, 2250, 2247, 2244, 2240, 2237, 2234, 2231, 2228, 2225, 2222, 2219, 2215, 2212, 2209, 2206, 2203, 2200, 2197, 2193, 2190, 2187, 2184, 2181, 2178, 2175, 2172, 2168, 2165, 2162, 2159, 2156, 2153, 2150, 2146, 2143, 2140, 2137, 2134, 2131, 2128, 2124, 2121, 2118, 2115, 2112, 2109, 2106, 2102, 2099, 2096, 2093, 2090, 2087, 2084, 2080, 2077, 2074, 2071, 2068, 2065, 2062, 2058, 2055, 2052, 2049, 2046, 2043, 2040, 2037, 2033, 2030, 2027, 2024, 2021, 2018, 2015, 2011, 2008, 2005, 2002, 1999, 1996, 1993, 1989, 1986, 1983, 1980, 1977, 1974, 1971, 1967, 1964, 1961, 1958, 1955, 1952, 1949, 1945, 1942, 1939, 1936, 1933, 1930, 1927, 1923, 1920, 1917, 1914, 1911, 1908, 1905, 1902, 1898, 1895, 1892, 1889, 1886, 1883, 1880, 1876, 1873, 1870, 1867, 1864, 1861, 1858, 1855, 1851, 1848, 1845, 1842, 1839, 1836, 1833, 1830, 1826, 1823, 1820, 1817, 1814, 1811, 1808, 1805, 1801, 1798, 1795, 1792, 1789, 1786, 1783, 1780, 1777, 1773, 1770, 1767, 1764, 1761, 1758, 1755, 1752, 1749, 1745, 1742, 1739, 1736, 1733, 1730, 1727, 1724, 1721, 1718, 1714, 1711, 1708, 1705, 1702, 1699, 1696, 1693, 1690, 1687, 1683, 1680, 1677, 1674, 1671, 1668, 1665, 1662, 1659, 1656, 1653, 1649, 1646, 1643, 1640, 1637, 1634, 1631, 1628, 1625, 1622, 1619, 1616, 1613, 1610, 1606, 1603, 1600, 1597, 1594, 1591, 1588, 1585, 1582, 1579, 1576, 1573, 1570, 1567, 1564, 1561, 1558, 1554, 1551, 1548, 1545, 1542, 1539, 1536, 1533, 1530, 1527, 1524, 1521, 1518, 1515, 1512, 1509, 1506, 1503, 1500, 1497, 1494, 1491, 1488, 1485, 1482, 1479, 1476, 1473, 1470, 1467, 1464, 1461, 1458, 1455, 1451, 1448, 1445, 1442, 1439, 1436, 1433, 1430, 1427, 1425, 1422, 1419, 1416, 1413, 1410, 1407, 1404, 1401, 1398, 1395, 1392, 1389, 1386, 1383, 1380, 1377, 1374, 1371, 1368, 1365, 1362, 1359, 1356, 1353, 1350, 1347, 1344, 1341, 1338, 1335, 1332, 1330, 1327, 1324, 1321, 1318, 1315, 1312, 1309, 1306, 1303, 1300, 1297, 1294, 1291, 1289, 1286, 1283, 1280, 1277, 1274, 1271, 1268, 1265, 1262, 1259, 1257, 1254, 1251, 1248, 1245, 1242, 1239, 1236, 1233, 1231, 1228, 1225, 1222, 1219, 1216, 1213, 1210, 1208, 1205, 1202, 1199, 1196, 1193, 1190, 1188, 1185, 1182, 1179, 1176, 1173, 1170, 1168, 1165, 1162, 1159, 1156, 1153, 1151, 1148, 1145, 1142, 1139, 1137, 1134, 1131, 1128, 1125, 1123, 1120, 1117, 1114, 1111, 1109, 1106, 1103, 1100, 1097, 1095, 1092, 1089, 1086, 1083, 1081, 1078, 1075, 1072, 1070, 1067, 1064, 1061, 1059, 1056, 1053, 1050, 1048, 1045, 1042, 1039, 1037, 1034, 1031, 1029, 1026, 1023, 1020, 1018, 1015, 1012, 1009, 1007, 1004, 1001, 999, 996, 993, 991, 988, 985, 983, 980, 977, 975, 972, 969, 966, 964, 961, 958, 956, 953, 951, 948, 945, 943, 940, 937, 935, 932, 929, 927, 924, 922, 919, 916, 914, 911, 908, 906, 903, 901, 898, 895, 893, 890, 888, 885, 882, 880, 877, 875, 872, 870, 867, 864, 862, 859, 857, 854, 852, 849, 847, 844, 841, 839, 836, 834, 831, 829, 826, 824, 821, 819, 816, 814, 811, 809, 806, 804, 801, 799, 796, 794, 791, 789, 786, 784, 781, 779, 776, 774, 772, 769, 767, 764, 762, 759, 757, 754, 752, 750, 747, 745, 742, 740, 737, 735, 733, 730, 728, 725, 723, 721, 718, 716, 713, 711, 709, 706, 704, 702, 699, 697, 694, 692, 690, 687, 685, 683, 680, 678, 676, 673, 671, 669, 666, 664, 662, 659, 657, 655, 653, 650, 648, 646, 643, 641, 639, 637, 634, 632, 630, 627, 625, 623, 621, 618, 616, 614, 612, 609, 607, 605, 603, 601, 598, 596, 594, 592, 589, 587, 585, 583, 581, 578, 576, 574, 572, 570, 568, 565, 563, 561, 559, 557, 555, 552, 550, 548, 546, 544, 542, 540, 538, 535, 533, 531, 529, 527, 525, 523, 521, 519, 516, 514, 512, 510, 508, 506, 504, 502, 500, 498,

496,494,492,490,488,486,484,482,480,478,475,473,471,469,467,465,
463,461,460,458,456,454,452,450,448,446,444,442,440,438,436,434,
432,430,428,426,424,423,421,419,417,415,413,411,409,407,405,404,
402,400,398,396,394,392,391,389,387,385,383,381,380,378,376,374,
372,371,369,367,365,363,362,360,358,356,354,353,351,349,347,346,
344,342,340,339,337,335,334,332,330,328,327,325,323,322,320,318,
317,315,313,312,310,308,307,305,303,302,300,298,297,295,293,292,
290,289,287,285,284,282,281,279,277,276,274,273,271,270,268,267,
265,263,262,260,259,257,256,254,253,251,250,248,247,245,244,242,
241,239,238,236,235,233,232,231,229,228,226,225,223,222,220,219,
218,216,215,213,212,211,209,208,207,205,204,202,201,200,198,197,
196,194,193,192,190,189,188,186,185,184,182,181,180,179,177,176,
175,174,172,171,170,168,167,166,165,164,162,161,160,159,157,156,
155,154,153,151,150,149,148,147,146,144,143,142,141,140,139,138,
136,135,134,133,132,131,130,129,128,126,125,124,123,122,121,120,
119,118,117,116,115,114,113,112,111,110,109,108,107,106,105,104,
103,102,101,100,99,98,97,96,95,94,93,92,91,90,89,88,
88,87,86,85,84,83,82,81,80,80,79,78,77,76,75,74,
74,73,72,71,70,69,69,68,67,66,65,65,64,63,62,62,
61,60,59,59,58,57,56,56,55,54,53,53,52,51,51,50,
49,49,48,47,47,46,45,45,44,43,43,42,41,41,40,40,
39,38,38,37,37,36,35,35,34,34,33,32,32,31,31,30,
30,29,29,28,28,27,27,26,26,25,25,24,24,23,23,22,
22,21,21,20,20,20,19,19,18,18,17,17,17,16,16,16,
15,15,14,14,14,13,13,13,12,12,12,11,11,11,10,10,
10,9,9,9,8,8,8,8,7,7,7,7,6,6,6,6,
5,5,5,5,4,4,4,4,4,3,3,3,3,3,3,
2,2,2,2,2,2,1,1,1,1,1,1,1,1,1,
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,
1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,
3,3,3,3,3,4,4,4,4,4,5,5,5,5,5,
6,6,6,6,7,7,7,7,8,8,8,9,9,9,9,10,
10,10,11,11,11,12,12,12,13,13,13,14,14,15,15,15,
16,16,16,17,17,18,18,19,19,19,20,20,21,21,22,22,
23,23,23,24,24,25,25,26,26,27,27,28,28,29,29,30,
31,31,32,32,33,33,34,34,35,36,36,37,37,38,39,39,
40,40,41,42,42,43,44,44,45,46,46,47,48,48,49,50,
50,51,52,52,53,54,55,55,56,57,57,58,59,60,60,61,
62,63,64,64,65,66,67,67,68,69,70,71,72,72,73,74,
75,76,77,77,78,79,80,81,82,83,83,84,85,86,87,88,
89,90,91,92,93,94,94,95,96,97,98,99,100,101,102,103,
104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,
121,122,123,124,125,126,127,128,129,130,131,132,134,135,136,137,
138,139,140,142,143,144,145,146,147,149,150,151,152,153,154,156,
157,158,159,160,162,163,164,165,167,168,169,170,172,173,174,175,
177,178,179,181,182,183,184,186,187,188,190,191,192,194,195,196,
198,199,200,202,203,204,206,207,209,210,211,213,214,216,217,218,
220,221,223,224,225,227,228,230,231,233,234,236,237,239,240,242,
243,244,246,247,249,250,252,253,255,257,258,260,261,263,264,266,
267,269,270,272,274,275,277,278,280,281,283,285,286,288,289,291,
293,294,296,298,299,301,302,304,306,307,309,311,312,314,316,317,
319,321,322,324,326,328,329,331,333,334,336,338,340,341,343,345,
347,348,350,352,354,355,357,359,361,362,364,366,368,370,371,373,
375,377,379,380,382,384,386,388,390,391,393,395,397,399,401,403,
405,406,408,410,412,414,416,418,420,422,423,425,427,429,431,433,
435,437,439,441,443,445,447,449,451,453,455,457,459,461,462,464,

Anexo Código

466,468,470,472,474,477,479,481,483,485,487,489,491,493,495,497,
499,501,503,505,507,509,511,513,515,518,520,522,524,526,528,530,
532,534,536,539,541,543,545,547,549,551,554,556,558,560,562,564,
566,569,571,573,575,577,580,582,584,586,588,591,593,595,597,599,
602,604,606,608,611,613,615,617,620,622,624,626,629,631,633,635,
638,640,642,645,647,649,651,654,656,658,661,663,665,668,670,672,
675,677,679,682,684,686,689,691,693,696,698,700,703,705,707,710,
712,715,717,719,722,724,727,729,731,734,736,739,741,743,746,748,
751,753,756,758,760,763,765,768,770,773,775,778,780,783,785,788,
790,793,795,797,800,802,805,807,810,812,815,817,820,823,825,828,
830,833,835,838,840,843,845,848,850,853,855,858,861,863,866,868,
871,873,876,879,881,884,886,889,892,894,897,899,902,905,907,910,
912,915,918,920,923,925,928,931,933,936,939,941,944,947,949,952,
955,957,960,962,965,968,970,973,976,979,981,984,987,989,992,995,
997,1000,1003,1005,1008,1011,1014,1016,1019,1022,1024,1027,1030,1033,1035,1038,
1041,1044,1046,1049,1052,1055,1057,1060,1063,1066,1068,1071,1074,1077,1079,1082,
1085,1088,1090,1093,1096,1099,1102,1104,1107,1110,1113,1116,1118,1121,1124,1127,
1130,1132,1135,1138,1141,1144,1146,1149,1152,1155,1158,1161,1163,1166,1169,1172,
1175,1178,1180,1183,1186,1189,1192,1195,1198,1200,1203,1206,1209,1212,1215,1218,
1220,1223,1226,1229,1232,1235,1238,1241,1244,1246,1249,1252,1255,1258,1261,1264,
1267,1270,1272,1275,1278,1281,1284,1287,1290,1293,1296,1299,1302,1305,1308,1310,
1313,1316,1319,1322,1325,1328,1331,1334,1337,1340,1343,1346,1349,1352,1355,1358,
1361,1363,1366,1369,1372,1375,1378,1381,1384,1387,1390,1393,1396,1399,1402,1405,
1408,1411,1414,1417,1420,1423,1426,1429,1432,1435,1438,1441,1444,1447,1450,1453,
1456,1459,1462,1465,1468,1471,1474,1477,1480,1483,1486,1489,1492,1495,1498,1501,
1504,1507,1510,1513,1516,1519,1523,1526,1529,1532,1535,1538,1541,1544,1547,1550,
1553,1556,1559,1562,1565,1568,1571,1574,1577,1580,1583,1587,1590,1593,1596,1599,
1602,1605,1608,1611,1614,1617,1620,1623,1626,1629,1633,1636,1639,1642,1645,1648,
1651,1654,1657,1660,1663,1666,1670,1673,1676,1679,1682,1685,1688,1691,1694,1697,
1700,1704,1707,1710,1713,1716,1719,1722,1725,1728,1731,1735,1738,1741,1744,1747,
1750,1753,1756,1759,1763,1766,1769,1772,1775,1778,1781,1784,1787,1791,1794,1797,
1800,1803,1806,1809,1812,1816,1819,1822,1825,1828,1831,1834,1837,1841,1844,1847,
1850,1853,1856,1859,1862,1866,1869,1872,1875,1878,1881,1884,1887,1891,1894,1897,
1900,1903,1906,1909,1913,1916,1919,1922,1925,1928,1931,1934,1938,1941,1944,1947,
1950,1953,1956,1960,1963,1966,1969,1972,1975,1978,1982,1985,1988,1991,1994,1997,
2000,2004,2007,2010,2013,2016,2019,2022,2026,2029,2032,2035,2038,2041,2044,2048};

```
volatile uint32_t ticks = 0;  
volatile uint32_t tick = 0;  
volatile uint32_t tickEnd = 0;
```

```
/* Private function prototypes -----*/
```

```
void    Configure_ADC(void);  
void    Activate_ADC(void);  
void    ConversionStartPoll_ADC_GrpRegular(void);  
void    LED_Init(void);  
void    LED_On(void);  
void    LED_Off(void);  
void    LED_Blinking(uint32_t Period);  
void    Configure_TIMTimeBase(void);
```

```
/* Extern functions -----*/
```

```
/* GPIO NANO homebrew function prototypes -----*/  
extern void    configurePin(GPIO_TypeDef *GPIOPORT, uint32_t GPIOPIN, uint32_t GPIOMODE,  
uint32_t PULL);
```

```

extern void          setPin(GPIO_TypeDef *GPIOx, uint32_t PinMask);
extern void          resetPin(GPIO_TypeDef *GPIOx, uint32_t PinMask);
extern void          statePin(GPIO_TypeDef *GPIOx, uint32_t PinMask, uint32_t
state);
extern void          togglePin(GPIO_TypeDef *GPIOx, uint32_t PinMask);
extern uint32_t      readPin(GPIO_TypeDef *GPIOx, uint32_t PinMask);
/* ADC NANO homebrew function prototypes -----*/
extern void          configureADC(void);
extern void          activateADC(void);
extern void          setupADC(void);
extern void          convertADC(void);
/* DAC NANO homebrew function prototypes -----*/
extern void          configureDAC(void);
extern void          activateDAC(void);
extern void          setupDAC(void);
extern void          writeDAC(uint16_t value);
/* DMA NANO homebrew function prototypes -----*/
extern void          configureDMA(uint32_t bufferName);
/* TIM NANO homebrew function prototypes -----*/
extern void          configureTIM(void);
extern void          configureClock(void);
/* EXTI NANO homebrew function prototypes -----*/
extern void          configureEXTI(void);
extern void          IRQHandler(void);
/* VCO NANO homebrew function prototypes -----*/
extern uint16_t      voltsToFrequency(uint16_t voltsOctave);
/* PWM NANO homebrew function prototypes -----*/
extern void          configurePWM(void);
extern void          configureDutyCycle(uint32_t channel, uint32_t D);

/* MAIN NANO homebrew function prototypes -----*/
void frequencyCalculations(void);
void testWaveform(uint16_t steps);
void testWavetable(void);
void tickDelay(uint32_t ticks);
uint32_t map(float in, float inMin, float inMax, float outMin, float outMax);

#define CLOCK      72000000

#define WAVEFORM          4
#define FREQUENCY        5

#define VDDA_APPLI                ((uint32_t)3300)
#define ADC_CONVERTED_DATA_BUFFER_SIZE ((uint32_t) 3)

/* Variable to report status of DMA transfer of ADC group regular conversions */
/* 0: DMA transfer is not completed */
/* 1: DMA transfer is completed */
/* 2: DMA transfer has not been started yet (initial state) */
uint8_t ubDmaTransferStatus = 2; /* Variable set into DMA interruption callback */

/* VCO NANO Variables -----*/
uint16_t voltsOctave, frequency, frequencyTest, sampleRate, wave, DACValue;
//ADC group regular conversion data
uint16_t convertedADC[ADC_CONVERTED_DATA_BUFFER_SIZE];
//ADC Readings

```

```

uint16_t A0,A1;
//Input Readings
uint16_t speedPin, speedState, wavePin, waveState;
//Delay Proccess
uint32_t LFODelay, timerARR = 10000;
//Sync Proccess
uint32_t oldTicks = 100, newTicks;
float periodSync, freqSync;

/* Main Code */

int main(void)
{
    /* Configure the system clock to 72 MHz */
    configureClock();

        /* Configure the timer time base */
        configureTIM();

        /* Configure the PWM timer */
        configurePWM();

        /* Configure the external interruptions base */
        configureEXTI();

        /* Initialize wave select input */
        configurePin(GPIOA, PIN10, INPUT, NOPULL);

        /* Initialize speed select input */
        configurePin(GPIOA, PIN11, INPUT, PULLUP);

        /* Configure DMA for data transfer from ADC */
        configureDMA((uint32_t)&convertedADC);

        /* Initialize A0 ADC - Waveform selector / A1 ADC - 1V/Oct input */
        setupADC();

        /* Initialize A2 DAC without trigger conversion */
        setupDAC();

    /* Infinite loop */
    while (1)
    {
        //Process Timer INTs
    }
}

void countDelay(uint32_t delayCounter){
    uint32_t counter;
    for (counter = 0 ; counter < delayCounter ; counter++) {
    }
}

void speedIRQHandler(void){
    //Button debounce
    countDelay(2000000);
}

```

```

    //3 Speeds Available
    if (speedState <= 1){ speedState++; } else { speedState = 0;}
}

void waveIRQHandler(void){
    //If the input is for changing the wave
    if (!INT_RESET){
        //Reset wave
        frecuenciaTest = 0;

        //Button debounce
        countDelay(1500000);

        //Reset wave
        frecuenciaTest = 0;

        //8 Waves Available
        if (waveState <= 8){ waveState++; } else { waveState = 0;}
    } else {
        //Calculate difference of ticks
        newTicks = ticks-oldTicks;

        //Calculate period
        periodSync = newTicks*0.0001;

        freqSync = 128/periodSync;

        //Update TIM2 INT frequency
        LL_TIM_SetAutoReload(TIM2, __LL_TIM_CALC_ARR(SystemCoreClock,
LL_TIM_GetPrescaler(TIM3), freqSync));

        //Reset wave
        frecuenciaTest = 0;

        //Update ticks variable
        oldTicks = ticks;
    }
}

void convertADC(void){

    LL_ADC_REG_StartConversion(ADC1);

    /* ADC RAW Readings */
    A0 = convertedADC[0];
    A1 = convertedADC[1];
}

//Function to map variables into another ranges
uint32_t map(float in, float inMin, float inMax, float outMin, float outMax) {
    // check it's within the range
    if (inMin<inMax) {
        if (in <= inMin)
            return outMin;
        if (in >= inMax)
            return outMax;
    } else { // cope with input range being backwards.

```

```

    if (in >= inMin)
        return outMin;
    if (in <= inMax)
        return outMax;
}
// calculate how far into the range we are
float scale = (in-inMin)/(inMax-inMin);
// calculate the output.
return outMin + scale*(outMax-outMin);
}

void AdcDmaTransferComplete_Callback()
{
    /* Update status variable of DMA transfer */
    ubDmaTransferStatus = 1;
}

/**
 * @brief This function handles DMA1 interrupt request.
 * @param None
 * @retval None
 */
void DMA1_Channel1_IRQHandler(void)
{
    /* Check whether DMA transfer complete caused the DMA interruption */
    if(LL_DMA_IsActiveFlag_TC1(DMA1) == 1)
    {
        /* Clear flag DMA global interrupt */
        /* (global interrupt flag: half transfer and transfer complete flags) */
        LL_DMA_ClearFlag_GI1(DMA1);

        /* Call interruption treatment function */
        AdcDmaTransferComplete_Callback();
    }
}

/**
 * @brief Timer update interrupt processing
 * @param None
 * @retval None
 */
void waveProcess(void)
{
    //DAC value selection
    if (frecuencyTest < 4095){
        //SINE WAVE
        if (waveState == 0) {DACValue =
sineTable[frecuencyTest];}
        //TRIANGLE WAVE
        else if (waveState == 1) {
            if (frecuencyTest < 2047) {DACValue += 2;}
            else if (frecuencyTest == 2047) { DACValue = 4095;}
            else {DACValue -= 2;}}
        //SQUARE WAVE
        else if (waveState == 2) {if (frecuencyTest < 2047) {DACValue = 4095;} else
{DACValue = 0;}}
        //SAW UP WAVE

```

```

        else if (waveState == 3) {DACValue = frequencyTest;}
        //SAW DOWN WAVE
        else if (waveState == 4) {DACValue = 4095-frequencyTest;}
        //STAIR UP WAVE
        else if (waveState == 5) {if (frequencyTest % 512 == 0) {DACValue =
frequencyTest;}}
        //STAIR DOWN WAVE
        else if (waveState == 6) {if (frequencyTest % 512 == 0) {DACValue = 4095-
frequencyTest;}}
        //STAIR UP & DOWN
        else if (waveState == 7) {
            if (frequencyTest == 0){ DACValue = 0;}
            if (frequencyTest % 256 == 0){
                if (frequencyTest < 2047) {DACValue += 511;}
                else if (frequencyTest == 2047) { DACValue = 4095;}
                else { DACValue -= 511;}}
        }
        //PULSE WAVE
        else if (waveState == 8) {if (frequencyTest < 255) {DACValue = 4095;} else
{DACValue = 0;}}
        //RANDOM WAVE
        else if (waveState == 9) {if (frequencyTest % 255 == 0){ DACValue =
LL_TIM_GetCounter(TIM3)*65;}}

        //Wave state counter
        frequencyTest++;}

        //Write voltage into DAC
        writeDAC(DACValue);

        //Map ADC values into the different velocities and update ADC readings
        if (speedState == 2){
            //HIGH SPEED
            timerARR = map(A0-2*A1,0,4095,50,1000);
            if (frequencyTest % 128 == 0){convertADC();}}
        else if (speedState == 1){
            //MEDIUM SPEED
            timerARR = map(A0-2*A1,0,4095,1000,8000);
            if (frequencyTest % 64 == 0){convertADC();}}
        else if (speedState == 0){
            //LOW SPEED
            timerARR = map(A0-2*A1,0,4095,7000,2000000);
            convertADC();}

        if (!INT_RESET){
            //Update TIM2 INT frequency
            LL_TIM_SetAutoReload(TIM2, timerARR);
        }

        //Prevent overflow
        if (frequencyTest >= 4095){
            frequencyTest = 0;}

        //Ticks counter
        ticks++;
    }

```

```

void TIM3_IRQHandler(void)
{
    /* Check whether CC1 interrupt is pending */
    if(LL_TIM_IsActiveFlag_CC1(TIM3) == 1)
    {
        /* Clear the update interrupt flag*/
        LL_TIM_ClearFlag_CC1(TIM3);
        if (waveState == 0) {configureDutyCycle(1, DACValue/2);}
        else if (waveState == 1) {configureDutyCycle(2, DACValue/6);}
        else if (waveState == 2) {configureDutyCycle(3, DACValue/3);}
        else if (waveState == 3) {configureDutyCycle(4, DACValue/6);}
        else if (waveState == 4) {configureDutyCycle(5, DACValue/5);}
        else if (waveState == 5) {configureDutyCycle(6, DACValue/4);}
        else if (waveState == 6) {configureDutyCycle(7, DACValue/2);}
        else if (waveState == 7) {configureDutyCycle(8, DACValue/3);}
        else if (waveState == 8) {configureDutyCycle(9, DACValue/6);}
        else if (waveState == 9) {configureDutyCycle(10, DACValue/6);}
    }
    /* Check whether CC2 interrupt is pending */
    if(LL_TIM_IsActiveFlag_CC2(TIM3) == 1)
    {
        /* Clear the update interrupt flag*/
        LL_TIM_ClearFlag_CC2(TIM3);
    }
    /* Check whether CC3 interrupt is pending */
    if(LL_TIM_IsActiveFlag_CC3(TIM3) == 1)
    {
        /* Clear the update interrupt flag*/
        LL_TIM_ClearFlag_CC3(TIM3);
    }
}

#ifdef USE_FULL_ASSERT

/**
 * @brief Reports the name of the source file and the source line number
 *        where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(char *file, uint32_t line)
{
    /* User can add his own implementation to report the file name and line number,
       ex: printf("Wrong parameters value: file %s on line %d", file, line) */

    /* Infinite loop */
    while (1)
    {
    }
}
#endif

```


3.3. ADSR

```

#include <Wire.h>
#include <Adafruit_MCP4725.h>

Adafruit_MCP4725 dac;

//A5 SCL
//A4 SDA

#define pulseHigh(pin) {digitalWrite(pin, HIGH); digitalWrite(pin, LOW); }

#define DEBUG    0

#define ATTACK  A3
#define DECAY   A0
#define SUSTAIN A2
#define RELEASE A1

// Pin definitions...
const int gatePin   = 2;
const int ledPin    = 3;
const int retrigPin = 4;
const int speedPin  = 5;

float alpha = 0.7; // this is the pole location (called below 'time constant') used in
the first-order difference equation
double alpha1 = 0.9; // initial value for attack
double alpha2 = 0.9; // initial value for decay
double alpha3 = 0.95; // initial value for release

float envelope = 0.0; // initialise the envelope
float CV0 = 0.0; // result of reads from potentiometers (yes - it will only be an
int, but helps with the casting!)
float CV1 = 0.0;
int CV2 = 0;
float CV3 = 0.0;

int drive = 0;
int sustain_Level= 0;
int scan = 0;

boolean note_active = false;
boolean retrig = false;
boolean trigger = false;
boolean noteRetrig = false;
boolean decay = false;
boolean release_done = true;
boolean gate = true;
boolean FASTENV = false;

void setup() {

  pinMode(gatePin, INPUT);
  pinMode(retrigPin, INPUT);
  pinMode(speedPin, INPUT);

```

```

pinMode(ledPin,    OUTPUT);

//Set gate interrupt
attachInterrupt(digitalPinToInterrupt(gatePin), gateTrigger, CHANGE);

//Enable serial communication if DEBUG mode is active
if (DEBUG){
  Serial.begin(9600);
}

//Set up DAC
dac.begin(0x60);
}

void loop() {

  //Read only one of the inputs each pass
  readADCs();
  //Produce envelope
  envelopeProcess();

}

void envelopeProcess(void){

  //If a note isn't active and we're triggered, then start one
  while(trigger){
    if(note_active == false){
      decay = false;
      //Drive toward full value
      drive = 4096;
      //Set 'time constant' alpha1 for attack phase
      alpha = alpha1;
      //Set the note active flag
      note_active = true;
      //We are not retriggered
      retrig = false;
    }
    //If we are not retriggered
    if (!retrig){
      //Update trigger status
      if (!digitalRead(retrigPin)){
        retrig = true;
      } else { retrig = false; }}

    //If retriggered, start new attack
    if (note_active && retrig && noteRetrig){
      alpha = alpha1;
      decay = false;
      drive = 4096;
    }

    //If attack reached the maximum value, start decay phase
    if((decay == false)&&(envelope > 4000)&&(drive == 4096)){
      //Set decay flag
      decay = true;
      //Drive toward sustain level

```

```

    drive = sustain_Level;
    //Set 'time constant' alpha2 for decay phase
    alpha=alpha2;
    retrig = false;
}

//Implement difference equation:  $y(k) = (1 - \alpha) * x(k) + \alpha * y(k-1)$ 
envelope = ((1.0-alpha)*drive+alpha*envelope);

//Write value to DAC
dac.setVoltage(envelope, false);

//Write value to LED
analogWrite(ledPin, envelope/16);
}

//If the decay has finished, switch to sustain phase
if(note_active == true){
    //Drive towards zero
    drive = 0;
    //Set time constant' alpha3 for release phase
    alpha = alpha3;
    //Turn off note_active flag
    note_active = false;
    //Set release flag false
    release_done = false;
}

//Implement the difference equation again (outside the while loop)
envelope = ((1.0-alpha)*drive+alpha*envelope);

//Write value to DAC
dac.setVoltage(envelope, false);

//Write value to LED
analogWrite(ledPin, envelope/16);

//Is the release phase ended?
if(envelope < 4){
    //Yes - so flag it
    release_done = true;
}
}

//Gate Interrption Routine
void gateTrigger(void){

    //Acquire gate state
    gate = !digitalRead(gatePin);

    // trigger an ADSR event from gate
    trigger = gate;

    //Watch gate
    noteRetrig = gate;

    //Fast envelope or not

```

```

FASTENV = digitalRead(speedPin);
}

//Read the input parameters
void readADCs(){

//Get the attack pole location
CV0=analogRead(ATTACK);

if (FASTENV){
//0.52 to 0.97
alpha1=0.98*cos(CV0/795);
alpha1=sqrt(alpha1);}
else {
//0.97 to 0.99999
alpha1=0.97+((1023-CV0)/35000.0);
alpha1=sqrt(alpha1);}

//Get the release pole location
CV1=analogRead(DECAY);

if (FASTENV){
alpha2=0.99*cos(CV1/795);
alpha2=sqrt(alpha2);}
else {
alpha2=0.97+((1023-CV1)/35000.0);
alpha2=sqrt(alpha2);}

//Get the sustain level
CV2=1023-analogRead(SUSTAIN);
sustain_Level=CV2<<2;

//Get the release pole location
CV3=analogRead(RELEASE);

if (FASTENV){
alpha3=0.99*cos(CV3/795);
alpha3=sqrt(alpha3);}
else {
alpha3=0.97+((1023-CV3)/35000.0);
alpha3=sqrt(alpha3);}

if (DEBUG){
Serial.println(alpha1);
}
}

```



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño e implementación de un sintetizador de audio modular basado en síntesis substractiva

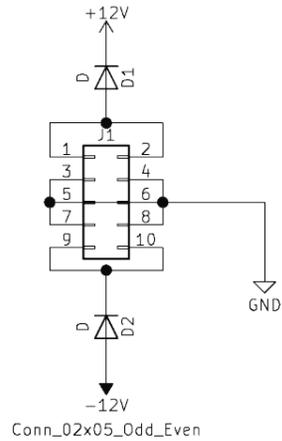
PLANOS - TFG

Grado en Ingeniería Electrónica Industrial y Automática

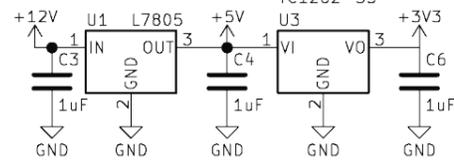
Autor: Gutiérrez-Ravé Olmos, Jorge

Tutor: Coll Arnau, Salvador

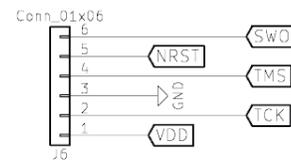
EURORACK POWER SUPPLY



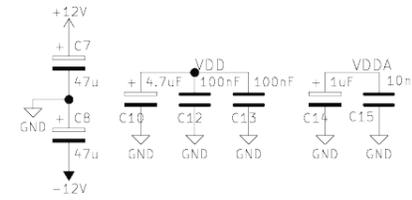
POWER STAGE



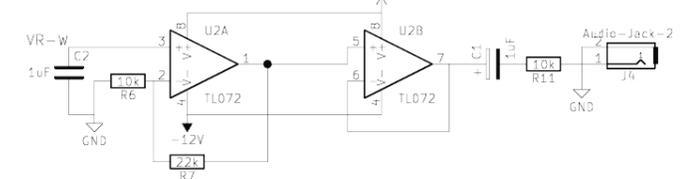
SWD MCU



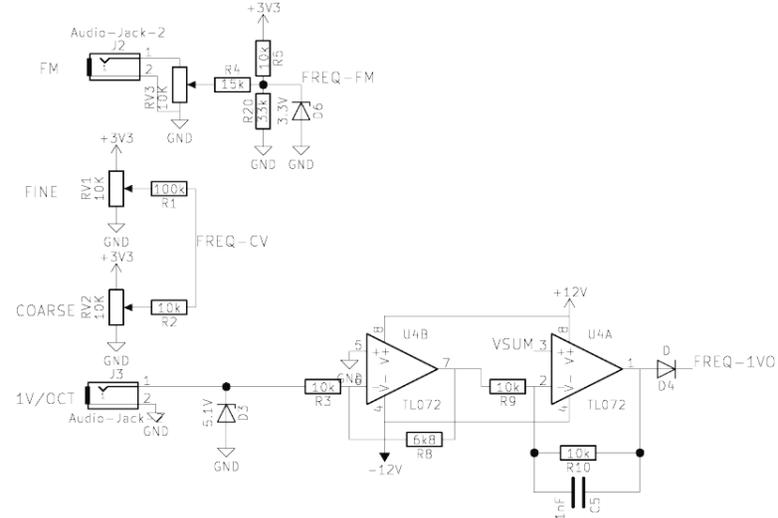
DECOUPLING



VCO OUTPUT STAGE



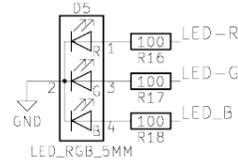
CV FREQUENCY



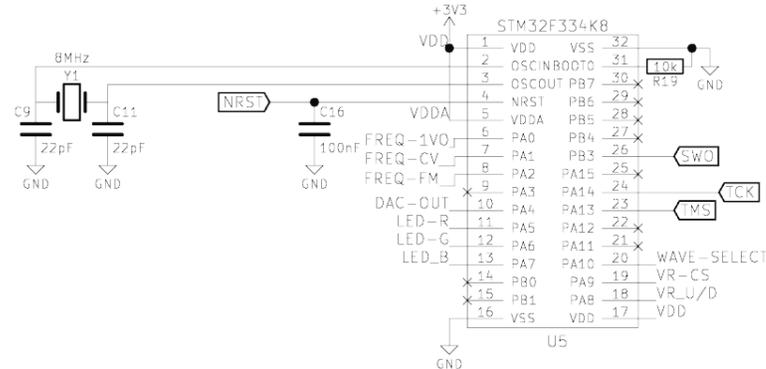
DIODE CLIP COMPENSATION



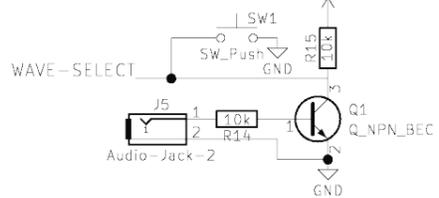
RGB LED



MCU

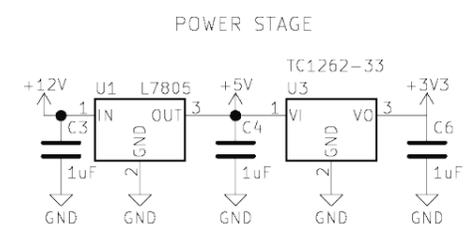
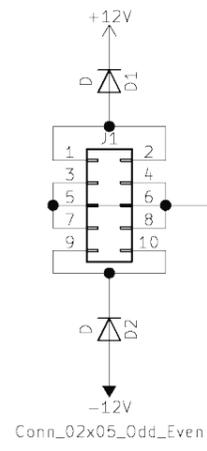


WAVE SELECTOR

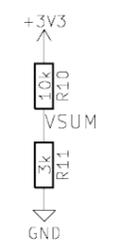


Referencia	Unidades	Valor	Ref. Vendedor	Descripción
C4,C2,C1	3	1uF	VJ1206Y105MXQTW1BC	Unpolarized capacitor
C5	1	1nF	VJ1206Y102MXQTW1BC	Unpolarized capacitor
C6,C7	2	47u	VE-470M1CTR-0505	Polarised capacitor
C8,C10	2	22pF	VJ1206A220KXQPW1BC	Unpolarized capacitor
C9	1	4.7uF	M865080640004	Polarised capacitor
C13,C3	2	1uF	M865060640001	Polarised capacitor
C14	1	10nF	VJ1206Y103MXQTW1BC	Unpolarized capacitor
C15,C11,C12	3	100nF	C1206C104K5RACTU	Unpolarized capacitor
D1,D2	2	D	1N4148W-7-F	Diode
D3	1	5.1V	BZT52C5V1-E3-08	Zener Diode
D4	1	3.3V	BZT52C3V3-E3-18	Zener Diode
D5	1	LED_RGB_5MM	WP154A4SUREQBFZGW	LED RGB, Common Cathode
J1	2	02x05	M20-9990546	Generic connector 02x05
J2,J3,J4,J5	4	Jack-2	TH350	Thonkiconn Jacks (PJ301M-12)
J6	1	Conn_01x06	M20-9770646	Generic connector 01x06
Q1	1	MMBT3904	MMBT3904	NPN Transistor
R2	1	100k	RC1206FR-07100KL	Resistor
R4	1	22k	RC1206FR-0722KL	Resistor
R5	1	15k	RC1206FR-0715KL	Resistor
R6	1	33k	RC1206FR-0733KL	Resistor
R8	1	6k8	RC1206FR-076K8L	Resistor
R10,R11,R17,R9,R3,R11,R12,R13,R7	9	10k	RC1206FR-0710KL	Resistor
R14,R15,R16	3	100	RC1206FR-07100L	Resistor
RV21,RV2,RV3	3	10K	TH3100POT	VERTICAL: B10K - 10K Linear - D shaft
SW1	1	SW_Push	TL1105KF160Q	Push button switch
U1,U3	2	TL072	TL072CDR	Dual Operational Amplifiers
U2	1	L7805	L7805CDT-TR	Voltage Regulator 5V,
U4	1	TC1262-33	TC1262-3.3VDBTR	Voltage Regulator 3.3V,
U5	1	STM32F334K8	STM32F334K8T6	Microcontroller
Y1	1	8MHz	LFXTAL003156Bulk	Two pin crystal
PCB	1	2 Layer	100x30mm	AIIPCB
PANEL	1	PCB	128x30mm	AIIPCB
KNOBS	3	1900h	THKNOBS	Black - Glow Pointer (D-Shaft)

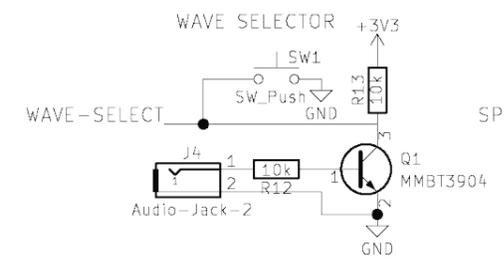
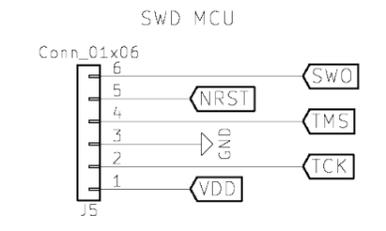
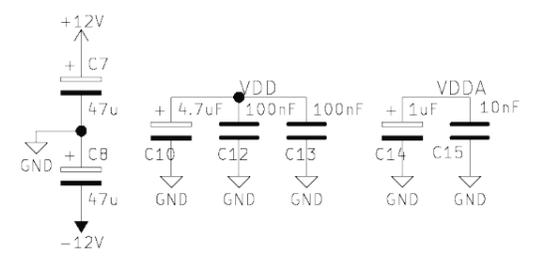
Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por: 13/05/2018	Gutiérrez-Ravé			
Revisado por: 15/05/2018	Coll Arnau			
Escala S/E	Esquemas electrónicos VCO.		Esquemas electrónicos.	Nº de Hoja 1.1



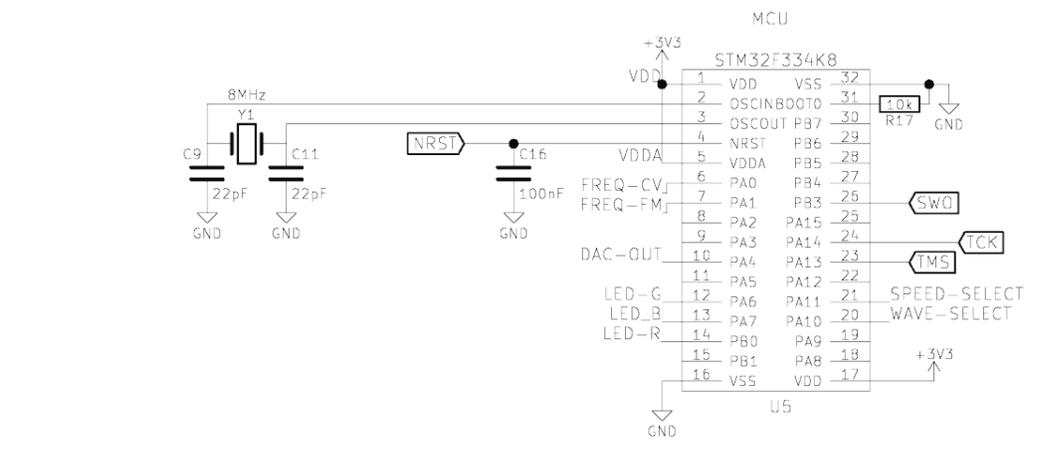
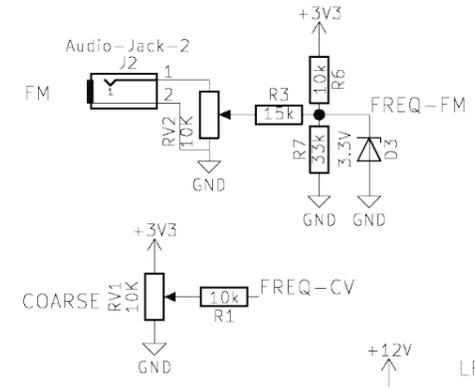
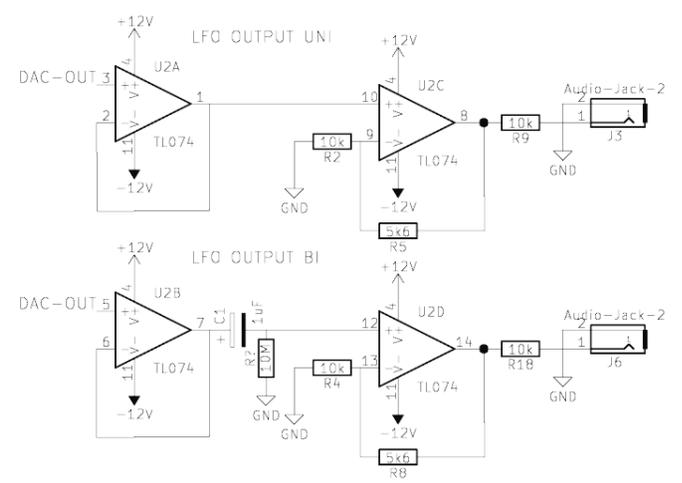
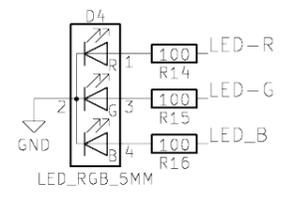
DIODE CLIP COMPENSATION



DECOUPLING

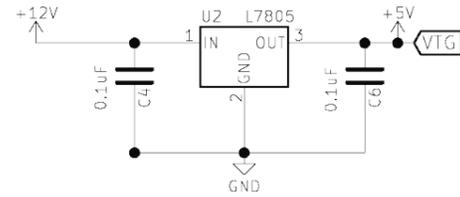
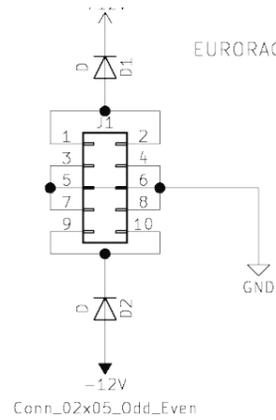


SPEED SELECTOR

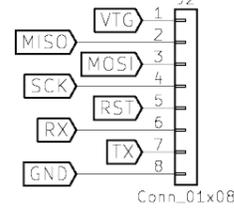


Component	Quantity	Value	Ref. Vendedor	Descripción
C1, C14	2	1uF	M865060640001	Polarised capacitor
C3, C4, C6	3	1uF	VJ1206Y105MXQTW1BC	Unpolarized capacitor
C7, C8	2	47u	VE-470M1CTR-0505	Polarised capacitor
C9, C11	2	22pF	VJ1206A220KXQPW1BC	Unpolarized capacitor
C10	1	4.7uF	M865080640004	Polarised capacitor
C12, C13, C16	3	100nF	C1206C104K5RACTU	Unpolarized capacitor
C15	1	10nF	VJ1206Y103MXQTW1BC	Unpolarized capacitor
D1, D2	2	D	1N4148W-7-F	Diode
D3	1	3.3V	BZT52C3V3-E3-18	Zener Diode
D4	1	LED_RGB_5MM	WP154A4SUREQBFGW	LED RGB, Common Cathode
J1	1	02x05	M20-9990546	Generic connector 02x05
J2, J3, J4, J6	4	Jack-2	TH350	Thonkiconn Jacks (PJ301M-12)
J5	1	Conn_01x06	M20-9770646	Generic connector 01x06
Q1	1	MMBT3904	MMBT3904	NPN Transistor
R1, R2, R4, R6, R9, R10, R12, R13, R17, R18	10	10k	RC1206FR-0710KL	Resistor
R3	1	15k	RC1206FR-0715KL	Resistor
R5, R8	2	5k6	RC1206FR-075K6L	Resistor
R7	1	33k	RC1206FR-0733KL	Resistor
R11	1	3k	RC1206FR-073KL	Resistor
R14, R15, R16	3	100	RC1206FR-07100L	Resistor
R19	1	10M	RC1206FR-0710ML	Resistor
RV1, RV2	2	10K	TH3100POT	B10K - 10K Linear - D shaft
SW1, SW2	2	SW_Push	TL1105KF160Q	Push button switch
U1	1	L7805	L7805CDT-TR	Voltage Regulator 5V,
U2	1	TL074	TL074CDR	Quad Operational Amplifiers
U3	1	TC1262-33	TC1262-3.3VDBTR	Voltage Regulator 3.3V,
U5	1	STM32F334K8	STM32F334K8T6	Microcontroller
Y1	1	8MHz	LFXTAL003156Bulk	Two pin crystal
PCB	1	2 Layer	100x30mm	AIIPCB
Referencia	Unidades	Valor	Ref. Vendedor	Descripción

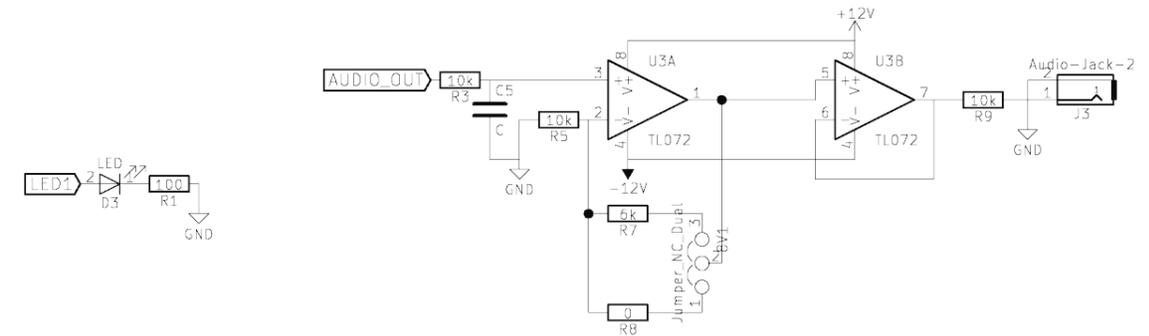
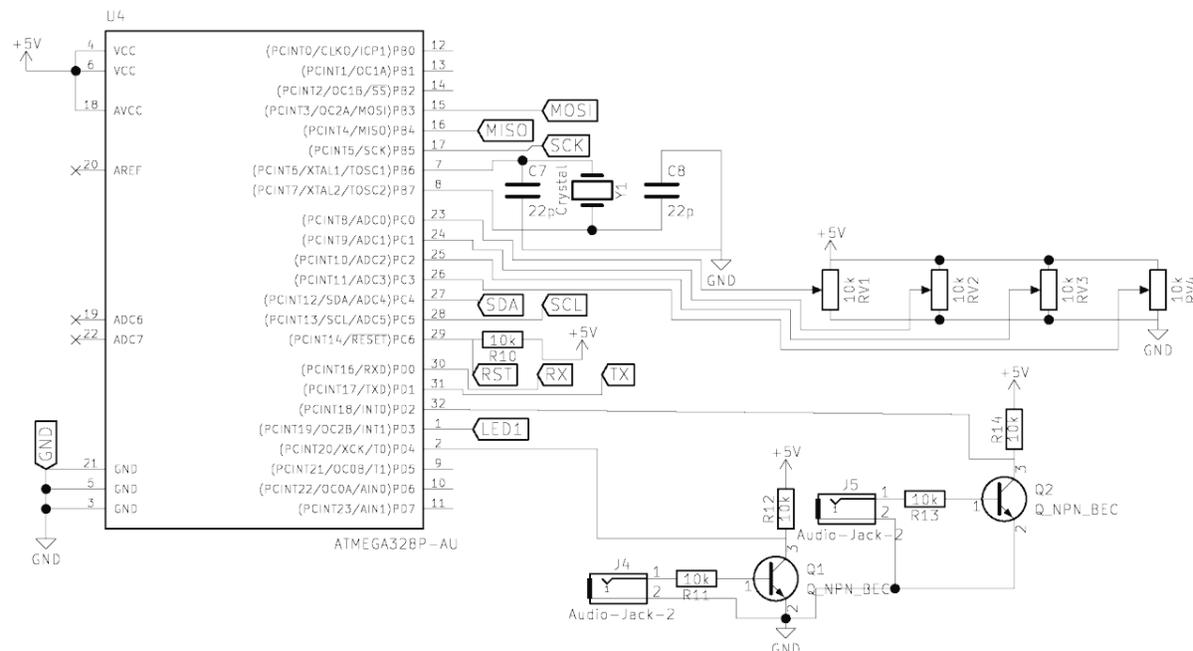
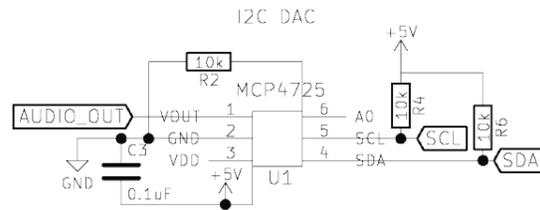
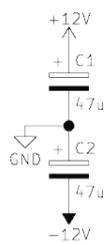
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	S/E			Esquemas electrónicos.	Nº de Hoja
	Esquemas electrónicos LFO.				1.2



PROGRAMMING CONNECTOR

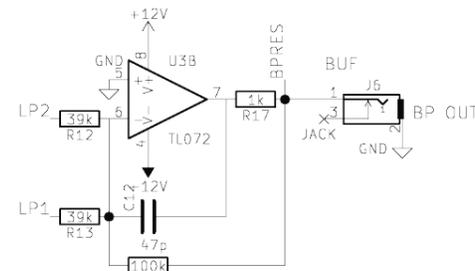
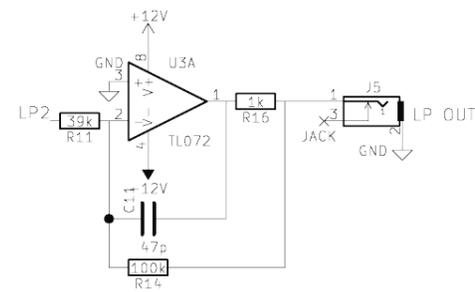
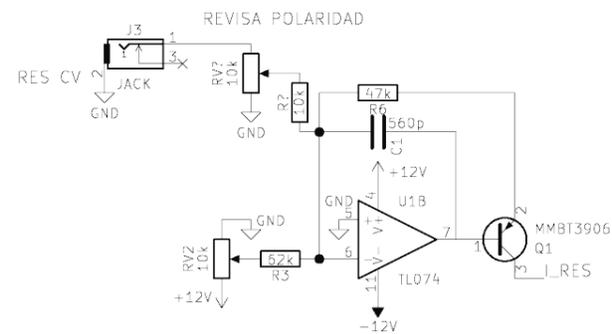
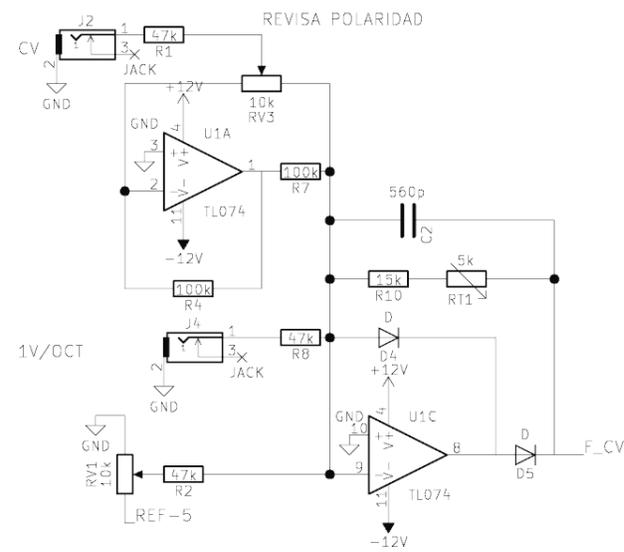
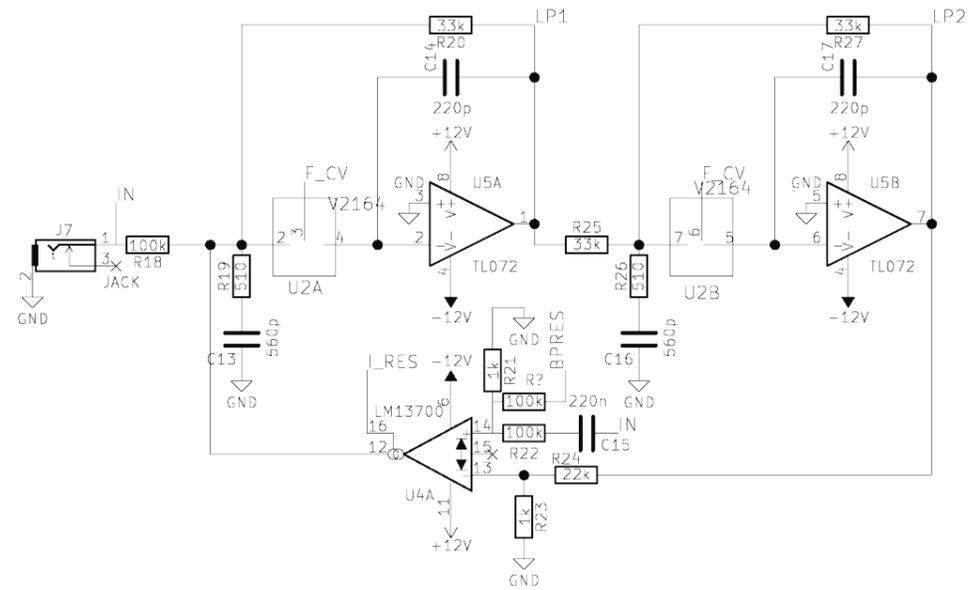


DECOUPLING

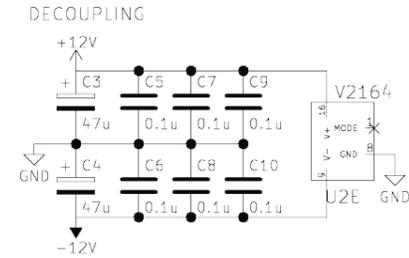
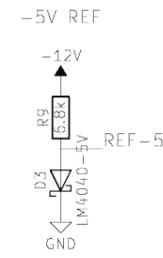
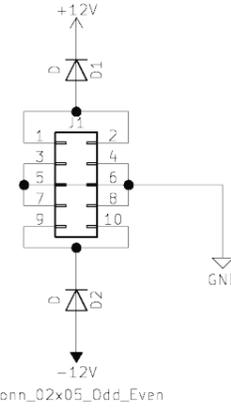


Component	Quantity	Value	Ref. Vendedor	Descripción
PCB	1	2 Layer	100x20mm	AIIPCB
PANEL	1	Acryl	128x30mm	AIIPCB
KNOBS	4	1900h	THKNOBS	Black - Glow Pointer (D-Shaft)
RV1 RV2 RV3 RV4	4	10k	TH3100POT	Alpha Potentiometer
J3 J4 J5	3	Audio-Jack-2	TH350	Thonkiconn Jacks (PJ301M-12)
C1 C2	2	47u	VE-470M1CTR-0505	Polarised Capacitor
C3 C4 C6	3	0.1uF	C1206C104K5RACTU	Unpolarised Capacitor
C7 C8	2	22p	VJ1206A220KXQPW1BC	Unpolarised Capacitor
D1 D2	2	D	1N4148W-7-F	Diode
D3	1	LED	LEDs:LED_D3.0mm	LED Diode
J1	1	Conn_02x05_Odd_Even	M20-9990546	Generic connector 02x05
J2	1	Conn_01x08	M20-9990534	Generic connector 01x08
Q1 Q2	2	Q_NPN_BEC	MMBT3904	Transistor NPN
R1	1	100	RC1206FR-07100L	Resistor
R10 R11 R12 R13 R14 R2 R3 R4 R5 R6 R9	11	10k	RC1206FR-07100KL	Resistor
R7	1	6k	RC1206FR-076KL	Resistor
R8	1	0	RC1206FR-070L	Resistor
U1	1	MCP4725	MCP4725	DAC Converter
U2	1	L7805	L7805CDT-TR	Linear Regulator
U3	1	TL072	TL072CDR	Dual Operational Amplifiers
U4	1	ATMEGA328P-AU	ATMEGA328P-AU	Microcontroller
Y1	1	Crystal	LFXTAL003156Bulk	Crystal
Referencia	Unidades	Valor	Ref. Vendedor	Descripción

Escala	Fecha	Nombre	Firmas	Esquemas electrónicos.	Nº de Hoja
S/E	13/05/2018	Gutiérrez-Ravé		Esquemas electrónicos ADSR.	13
	15/05/2018	Coll Arnau			
TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.				UNIVERSITAT POLITÈCNICA DE VALÈNCIA	

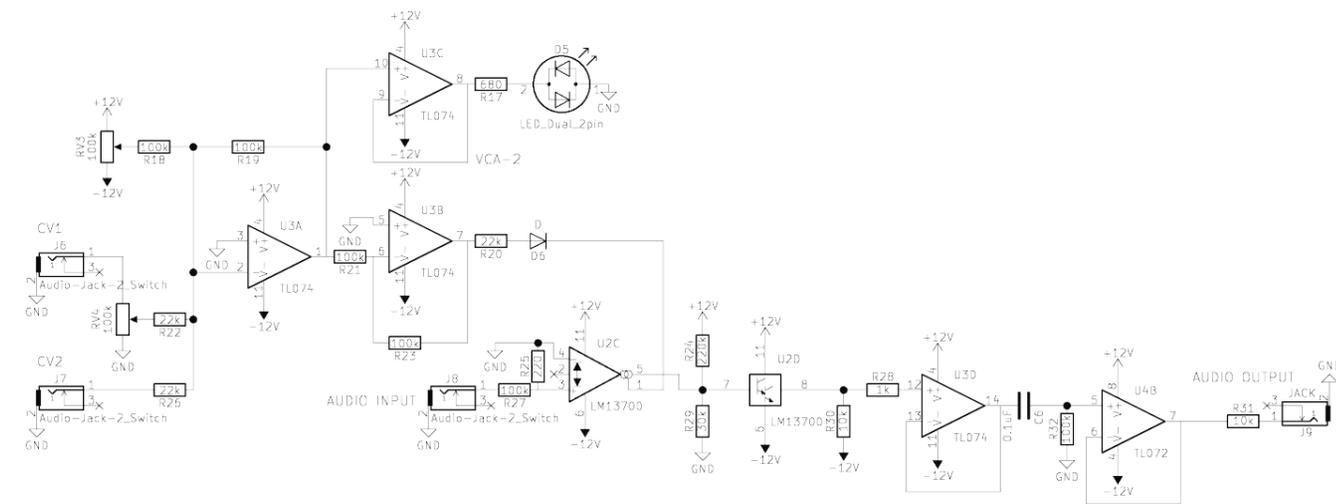
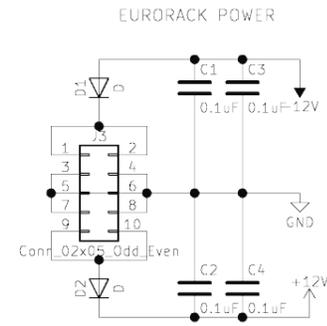
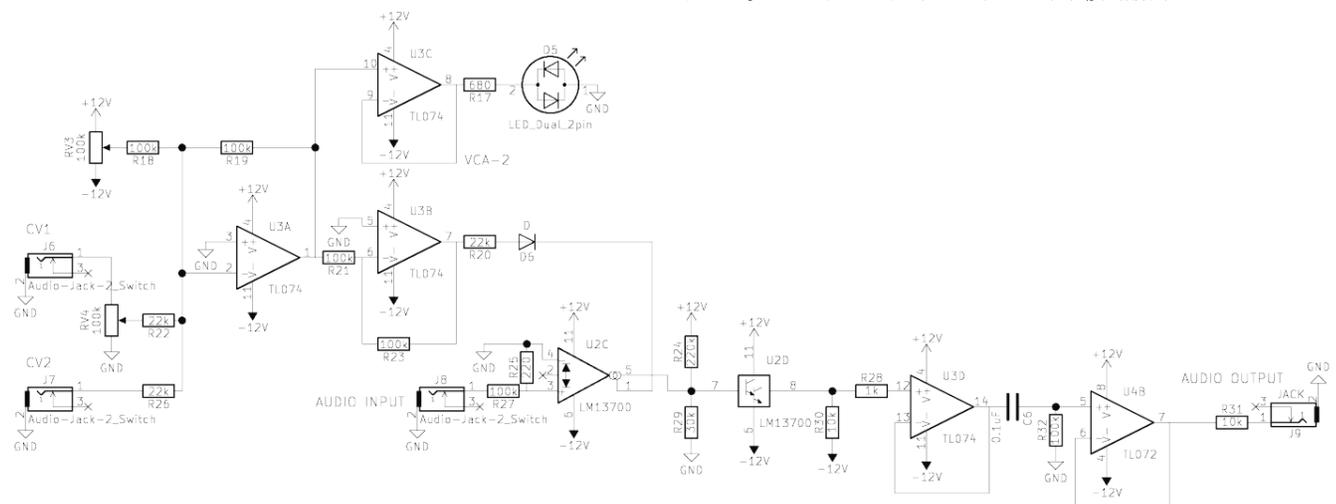


EURORACK POWER SUPPLY



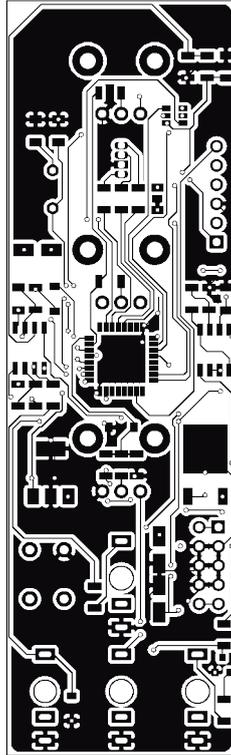
PCB	1	2 Layer	100x20mm	AIIPCB
PANEL	1	Acryl	128x30mm	AIIPCB
KNOBS	4	1900h	THKNOBS	Black - Glow Pointer (D-Shaft)
RV1 RV2 RV3	3	10k	TH3100POT	Alpha Potentiometer
U2	1	V2164	SOIC-16_3.9x9.9mm_Pitch1.27mm	VCA IC
J2 J3 J4 J5 J6 J7	6	JACK	TH350	Thonkiconn Jacks (PJ301M-12)
C11 C12	2	47p	VJ1206A470XQPW1BC	Unpolarised Capacitor
C14 C17	2	220p	VJ1206Y221KXACW1BC	Unpolarised Capacitor
C15	1	220n	VJ1206Y224KXJTW1BC	Unpolarised Capacitor
C1 C13 C16 C2	4	560p	VJ1206Y561KXAPW1BC	Unpolarised Capacitor
C3 C4	2	47u	VE-470M1CTR-0505	Polarised Capacitor
C10 C5 C6 C7 C8 C9	6	0.1u	C1206C104K5RACTU	Unpolarised Capacitor
D1 D2 D4 D5	4	D	1N4148W-7-F	Diode
D3	1	LM4040-5V	LM4040DYM3-5.0-TR	Voltage Reference Diode
J1	1	Conn_02x05_Odd_Even	M20-9990546	Generic connector 02x05
Q1	1	MMBT3906	MMBT3906LT3G	Transistor PNP
R1 R14 R15 R18 R4 R7 R8 R9 R22	9	100k	RC1206FR-07100KL	Resistor
R10	1	15k	RC1206FR-0715KL	Resistor
R11 R12 R13	3	39k	RC1206FR-0739KL	Resistor
R19 R26	2	510	RC1206FR-07510L	Resistor
R2 R6	2	47k	RC1206FR-0747KL	Resistor
R20 R25 R27	3	33k	RC1206FR-0733KL	Resistor
R16 R17 R21 R23	4	1k	RC1206FR-071KL	Resistor
R3	1	62k	RC1206FR-0762KL	Resistor
R24 R5	2	16k	RC1206FR-0716KL	Resistor
RT1	1	5k	PT6KV-502A2020	Trimmer
U1	1	TL074	TL074CDR	Quad Operational Amplifiers
U4	1	LM13700	LM13700MX/NOPB	Dual Transconductance Amplifiers
U3 U5	2	TL072	TL072CDR	Dual Operational Amplifiers
Referencia	Unidades	Valor	Ref. Vendedor	Descripción

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	Esquemas electrónicos VCF.			Esquemas electrónicos.	Nº de Hoja
S/E					1.4



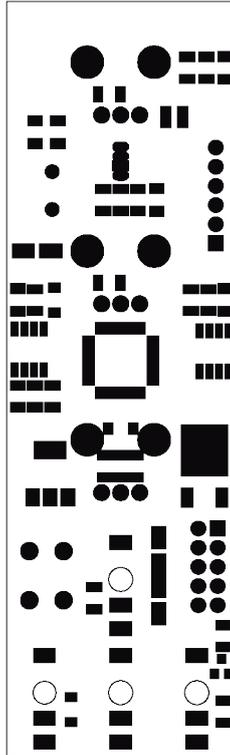
PCB	1	2 Layer	100x30mm	AIIPCB
PANEL	1	Acryl	128x30mm	AIIPCB
KNOBS	2	1900h	THKNOBS	Black - Glow Pointer (D-Shaft)
RV1 RV3	2	100k	TH3100POT	Alpha Potentiometer
RV2 RV4	2	100k	TH2100POT	Song Huei Potentiometer
J1 J2 J4 J5 J6 J7 J8 J9	8	Jack-2	TH350	Thonkiconn Jacks (PJ301M-12)
C1 C2 C3 C4	4	0.1uF	C1206C104K5RACTU	Unpolarized capacitor
D1 D2	2	D	1N4148W-7-F	Diode
D3 D4	2	LED	BZT52C5V1-E3-08	LED Diode
J3	2	02x05	M20-9990546	Generic connector 02x05
R1 R12 R15 R18 R19 R21 R4 R5 R6 R8	10	100k	RC1206FR-07100KL	Resistor
R10 R23	2	220k	RC1206FR-07220KL	Resistor
R11 R24	2	30k	RC1206FR-0730KL	Resistor
R16 R25	2	10k	RC1206FR-0710KL	Resistor
R17 R26	2	1k	RC1206FR-071KL	Resistor
R13 R14 R2 R20 R3 R7	6	22k	RC1206FR-0722KL	Resistor
R22 R9	2	220	RC1206FR-07220L	Resistor
U1 U3	2	TL072	TL072CDR	Dual Operational Amplifiers
U2	1	LM13700	LM13700MX/NOPB	Dual Transconductance Amplifiers
Referencia	Unidades	Valor	Ref. Vendedor	Descripción

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	Esquemas electrónicos VCA.			Esquemas electrónicos.	Nº de Hoja 15

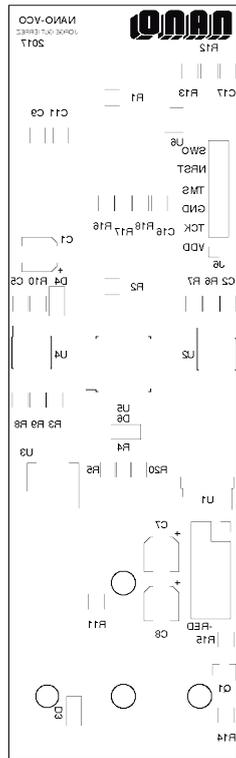


1	Circuito impreso VCO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÀCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	Capa de cobre trasera VCO. Alzado, planta y perfil.			Circuito impreso VCO.	Nº de Hoja 2.1.1
1:1					



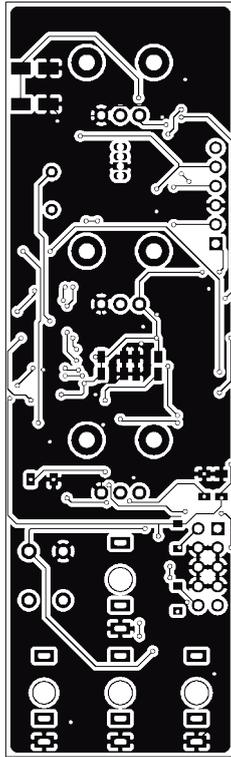
1	Circuito impreso VCO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad
	Fecha	Nombre	Firmas		
Diseñado por:	13/05/2018	Gutiérrez-Ravé		 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	
Revisado por:	15/05/2018	Coll Arnau			
Escala	Capa de máscara trasera VCO. Alzado, planta y perfil.			Circuito impreso VCO.	Nº de Hoja 2.1.2
1:1					



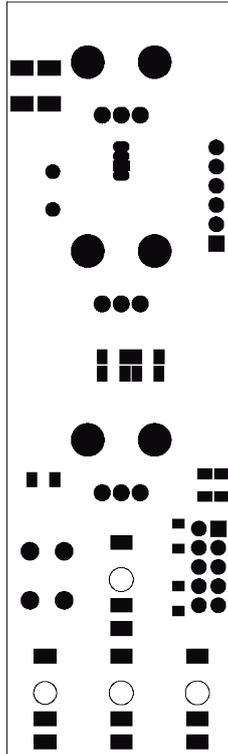
1	Circuito impreso VCO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala	Capa de serigrafía trasera VCO. Alzado, planta y perfil.	Circuito impreso VCO.	Nº de Hoja
1:1			2.1.3



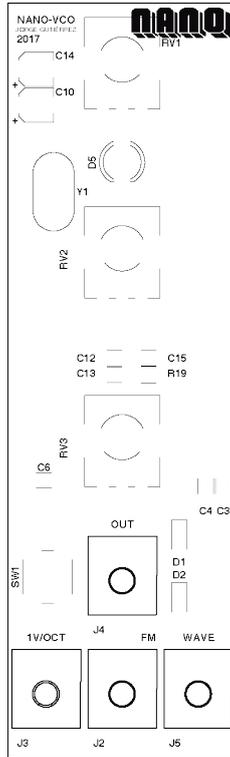
1	Circuito impreso VCO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad
	Fecha	Nombre	Firmas		
Diseñado por:	13/05/2018	Gutiérrez-Ravé			TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.
Revisado por:	15/05/2018	Coll Arnau			
				 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	
Escala	Capa de cobre delantera VCO. Alzado, planta y perfil.		Circuito impreso VCO.	Nº de Hoja	
1:1				2.14	



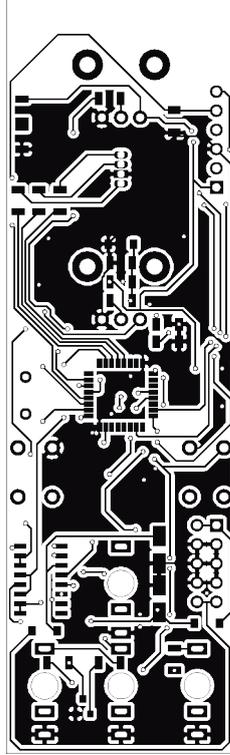
1	Circuito impreso VCO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala	Capa de máscara delantera VCO. Alzado, planta y perfil.	Circuito impreso VCO.	Nº de Hoja
1:1			2.15

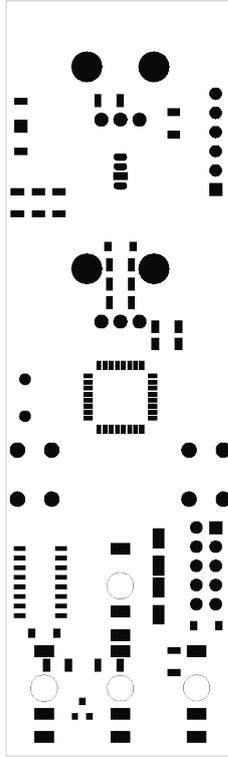


1	Circuito impreso VCO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad
	Fecha	Nombre	Firmas		
Diseñado por:	13/05/2018	Gutiérrez-Ravé	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.		 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Revisado por:	15/05/2018	Coll Arnau			
Escala 1:1	Capa de serigrafía delantera VCO. Alzado, planta y perfil.		Circuito impreso VCO.	Nº de Hoja 2.1.6	



1	Circuito impreso LFO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

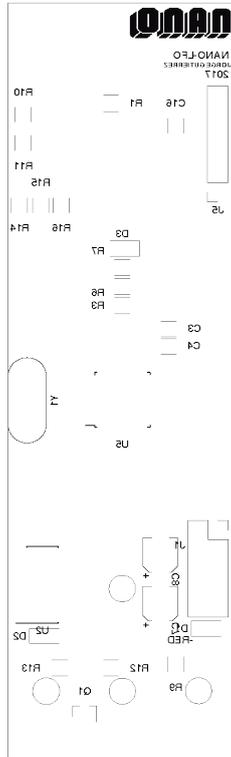
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	Capa de cobre trasera LFO. Alzado, planta y perfil.			Circuito impreso LFO.	Nº de Hoja 2.2.1
1:1					



1	Circuito impreso LFO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

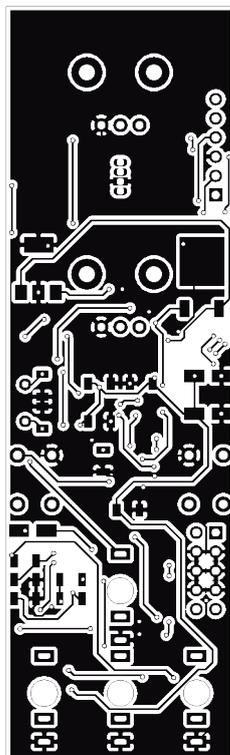
Escala 1:1	Capa de máscara trasera LFO. Alzado, planta y perfil.	Circuito impreso LFO.	Nº de Hoja 2.2.2
----------------------	--	-----------------------	----------------------------



1	Circuito impreso LFO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

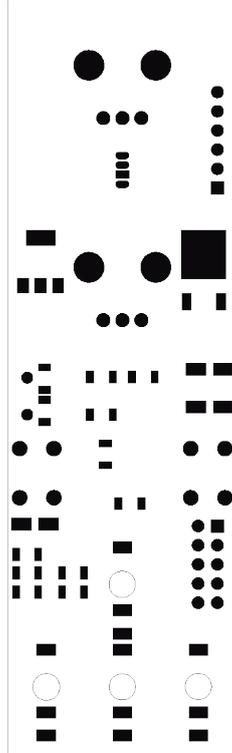
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala	Capa de serigrafía trasera LFO. Alzado, planta y perfil.	Circuito impreso LFO.	Nº de Hoja
1:1			2.2.3



1	Circuito impreso LFO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

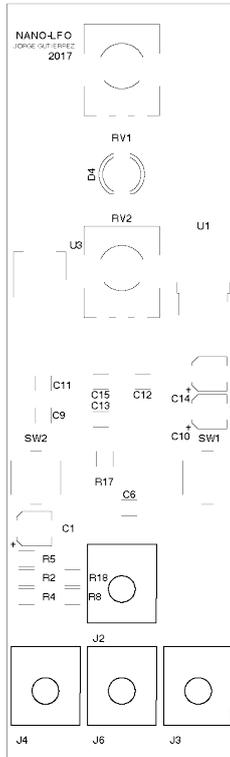
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	Capa de cobre delantera LFO. Alzado, planta y perfil.			Circuito impreso LFO.	Nº de Hoja 2.2.4
1:1					



1	Circuito impreso LFO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

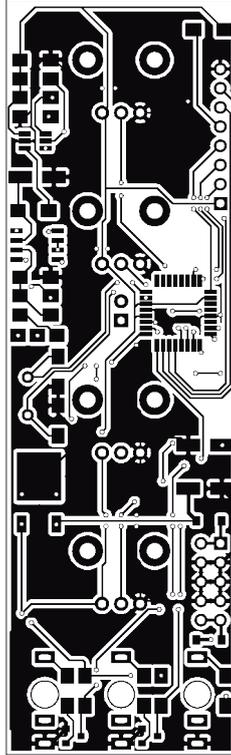
Escala	Capa de máscara delantera LFO. Alzado, planta y perfil.	Circuito impreso LFO.	Nº de Hoja
1:1			2.2.5



1	Circuito impreso LFO	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

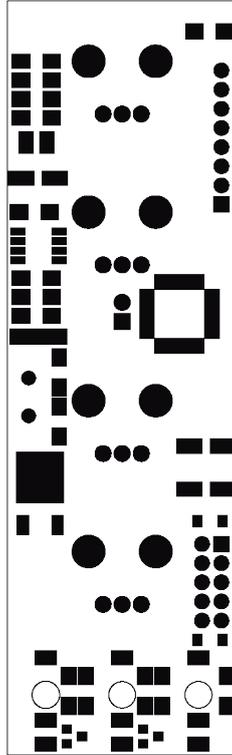
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala	Capa de serigrafía delantera LFO. Alzado, planta y perfil.	Circuito impreso LFO.	Nº de Hoja
1:1			2.2.6



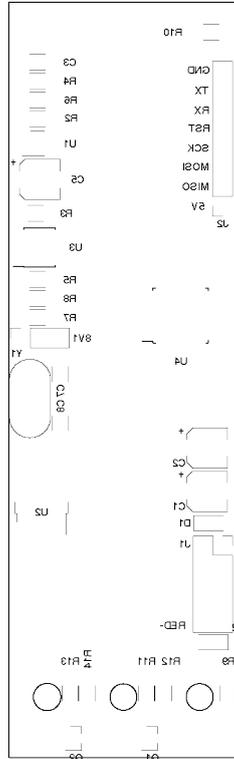
1	Circuito impreso ADSR	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala 1:1	Capa de cobre trasera ADSR. Alzado, planta y perfil.			Circuito impreso ADSR.	Nº de Hoja 2.3.1



1	Circuito impreso ADSR	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad
	Fecha	Nombre	Firmas		
Diseñado por:	13/05/2018	Gutiérrez-Ravé			 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Revisado por:	15/05/2018	Coll Arnau			
Escala 1:1	Capa de máscara trasera ADSR. Alzado, planta y perfil.		Circuito impreso ADSR.	Nº de Hoja 2.3.2	

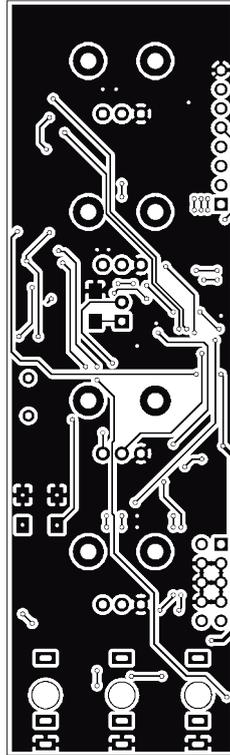
TFG.
 Diseño e Implementación de un
 sintetizador de audio modular
 basado en síntesis substractiva.



1	Circuito impreso ADSR	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis subtractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

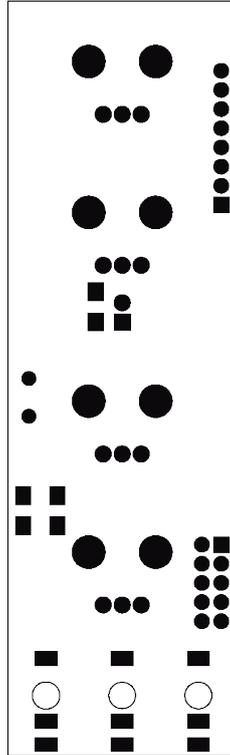
Escala	Capa de serigrafía trasera ADSR. Alzado, planta y perfil.	Circuito impreso ADSR.	Nº de Hoja
1:1			2.3.3



1	Circuito impreso ADSR	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÀCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

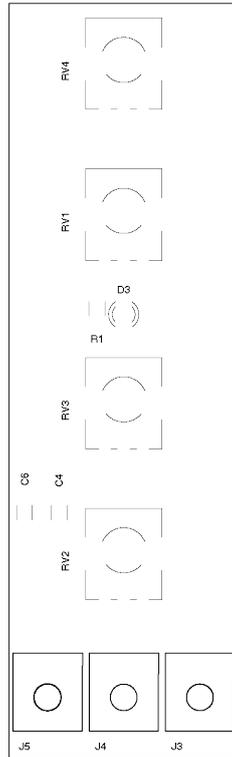
Escala	Capa de cobre delantera ADSR. Alzado, planta y perfil.	Circuito impreso ADSR.	Nº de Hoja
1:1			2.3.4



1	Circuito impreso ADSR	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

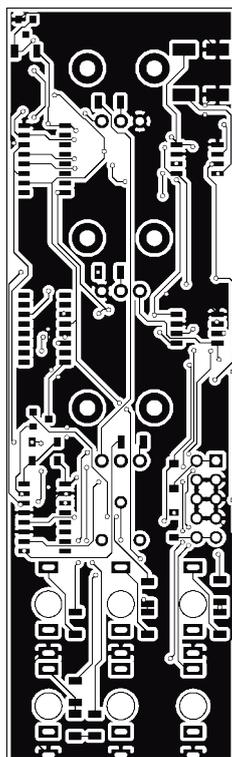
Escala	Capa de máscara delantera ADSR. Alzado, planta y perfil.	Circuito impreso ADSR.	Nº de Hoja
1:1			2.3.5



1	Circuito impreso ADSR	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

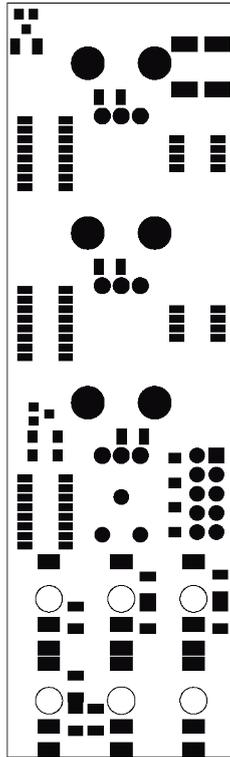
Escala	Capa de serigrafía delantera ADSR. Alzado, planta y perfil.	Circuito impreso ADSR.	Nº de Hoja
1:1			2.3.6



1	Circuito impreso VCF	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

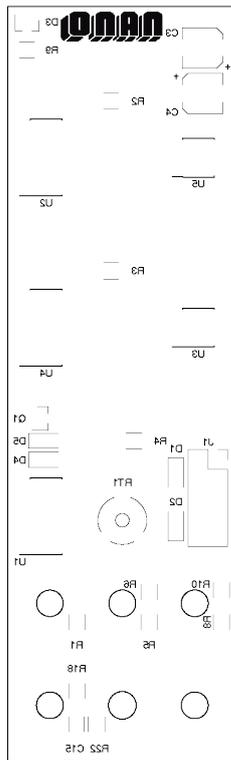
Escala 1:1	Capa de cobre trasera VCF. Alzado, planta y perfil.	Circuito impreso VCF.	Nº de Hoja 2.4.1
-------------------	--	--------------------------	-------------------------



1	Circuito impreso VCF	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

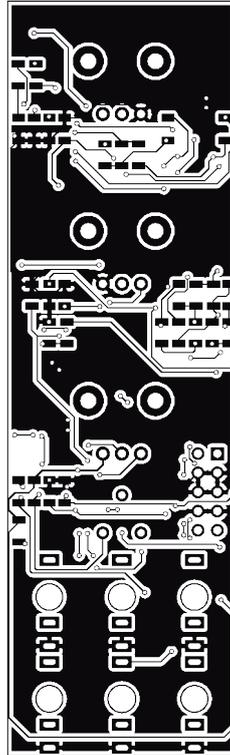
Escala 1:1	Capa de máscara trasera VCF. Alzado, planta y perfil.	Circuito impreso VCF.	Nº de Hoja 2.4.2
-------------------	--	--------------------------	-------------------------



1	Circuito impreso VCF	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

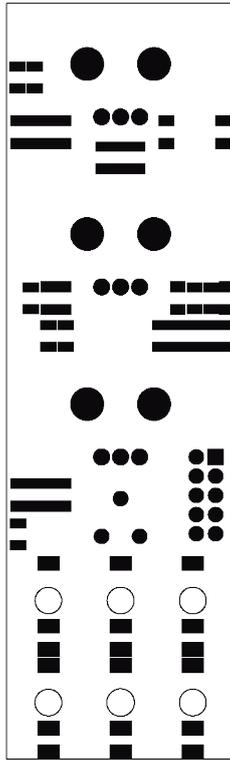
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala	Capa de serigrafía trasera VCF. Alzado, planta y perfil.	Circuito impreso VCF.	Nº de Hoja
1:1			2.4.3



1	Circuito impreso VCF	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

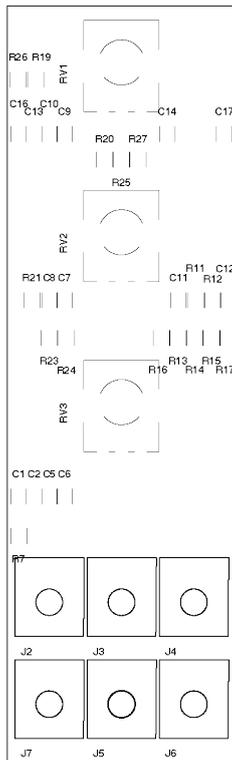
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	Capa de cobre delantera VCF. Alzado, planta y perfil.			Circuito impreso VCF.	Nº de Hoja 2.4.4
1:1					



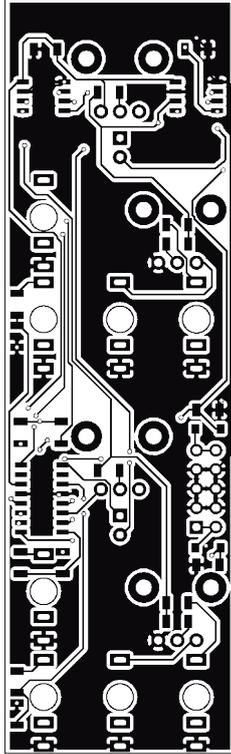
1	Circuito impreso VCF	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala 1:1	Capa de máscara delantera VCF. Alzado, planta y perfil.	Circuito impreso VCF.	Nº de Hoja 2.4.5
--------------------------	--	--------------------------	--------------------------------

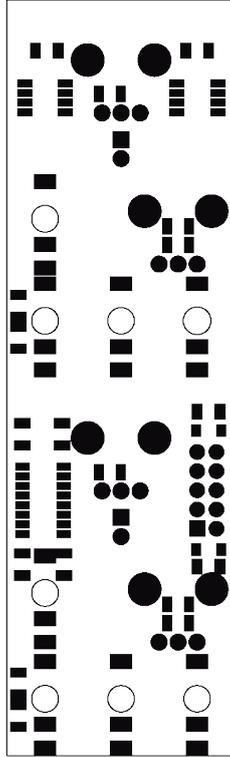


1	Circuito impreso VCF	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad
	Fecha	Nombre	Firmas		
Diseñado por:	13/05/2018	Gutiérrez-Ravé		 UNIVERSITAT POLITÈCNICA DE VALÈNCIA	
Revisado por:	15/05/2018	Coll Arnau			
Escala	Capa de serigrafía delantera VCF. Alzado, planta y perfil.		Circuito impreso VCF.	Nº de Hoja 2.4.6	
1:1					



1	Circuito impreso VCA	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

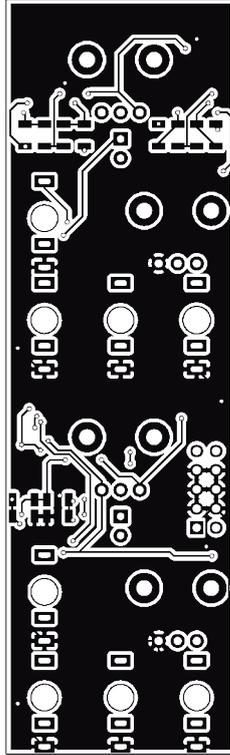
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala 1:1	Capa de cobre trasera VCA. Alzado, planta y perfil.			Circuito impreso VCA.	Nº de Hoja 2.5.1



1	Circuito impreso VCA	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

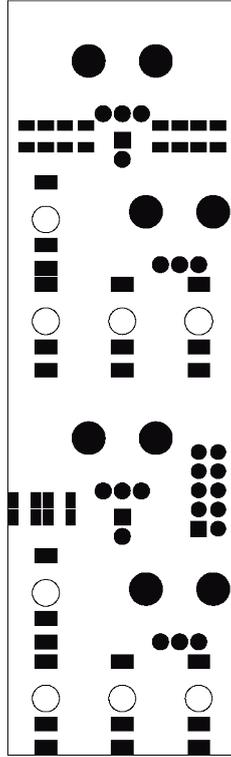
Escala	Capa de máscara trasera VCA. Alzado, planta y perfil.	Circuito impreso VCA.	Nº de Hoja
1:1			2.5.2



1	Circuito impreso VCA	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

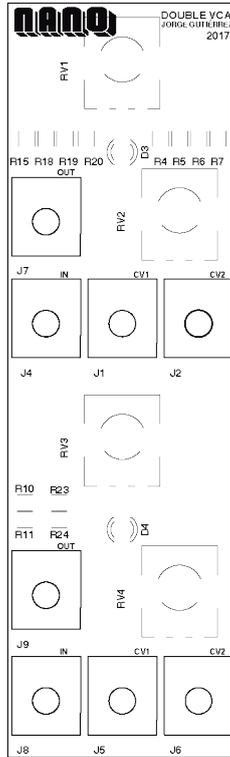
Escala	Capa de cobre delantera VCA. Alzado, planta y perfil.	Circuito impreso VCA.	Nº de Hoja
1:1			2.5.4



1	Circuito impreso VCA	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

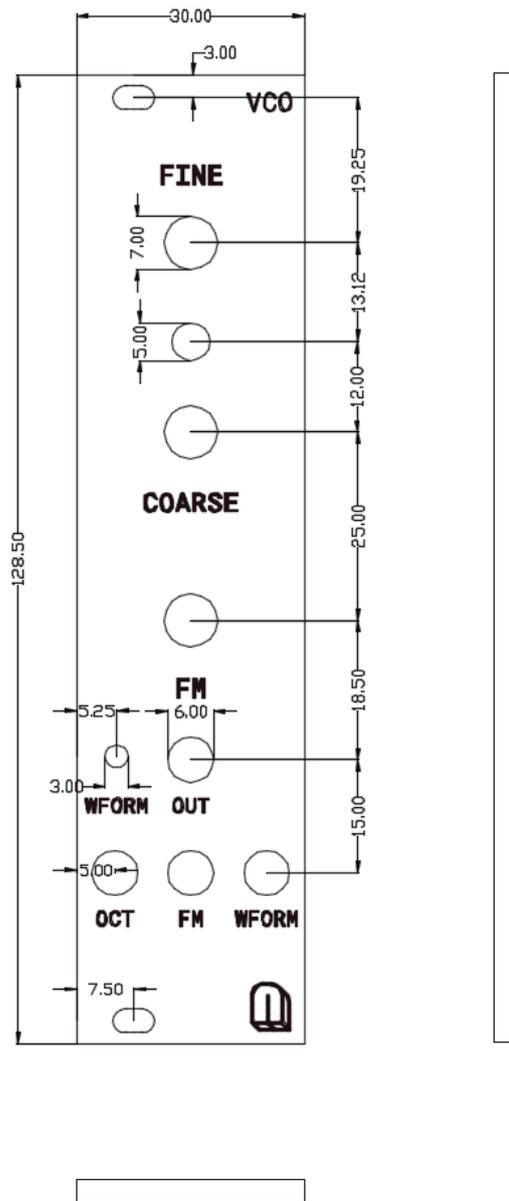
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala	Capa de máscara delantera VCA. Alzado, planta y perfil.	Circuito impreso VCA.	Nº de Hoja
1:1			2.5.5



1	Circuito impreso VCA	100x30x1,6 mm	FR4 1.6mm	#010	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

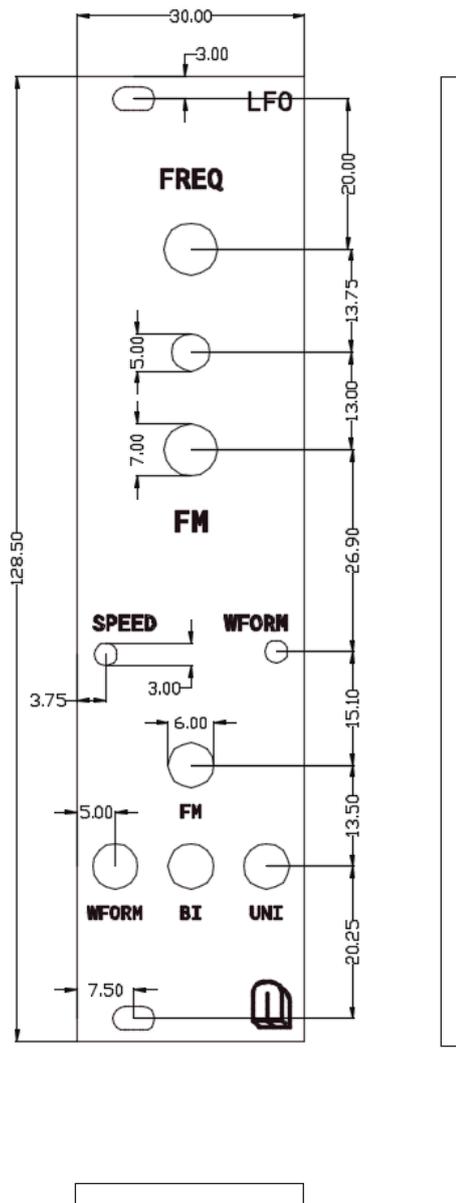
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	Capa de serigrafía delantera VCA. Alzado, planta y perfil.			Circuito impreso VCA.	Nº de Hoja 2.5.6
1:1					



1	Panel VCO	128,5x30x3 mm	MDF 3mm	#005	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

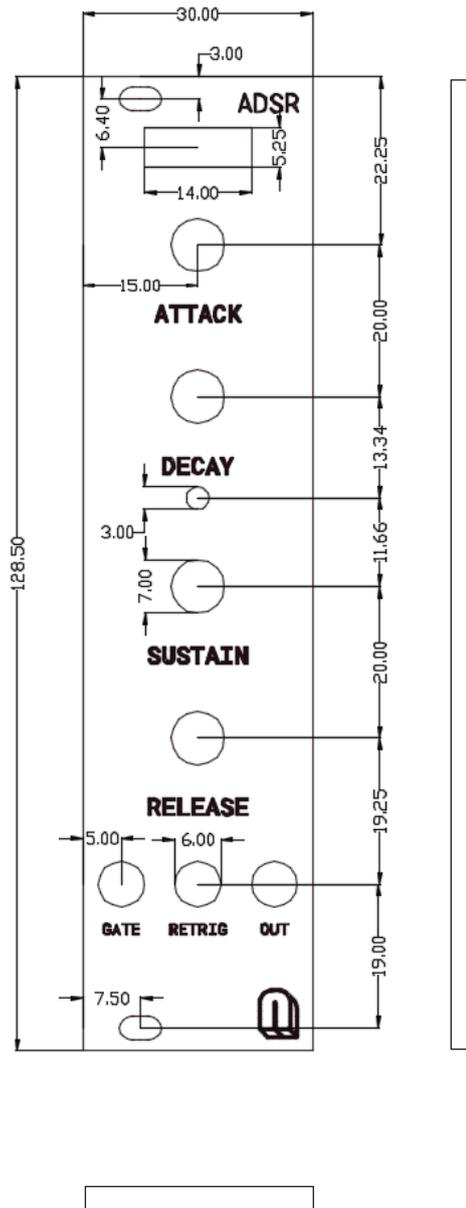
Escala	Panel VCO. Alzado, planta y perfil.	Componentes Mecánicos.	Nº de Hoja
1:1			3.1



1	Panel LFO	128,5x30x3 mm	MDF 3mm	#004	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

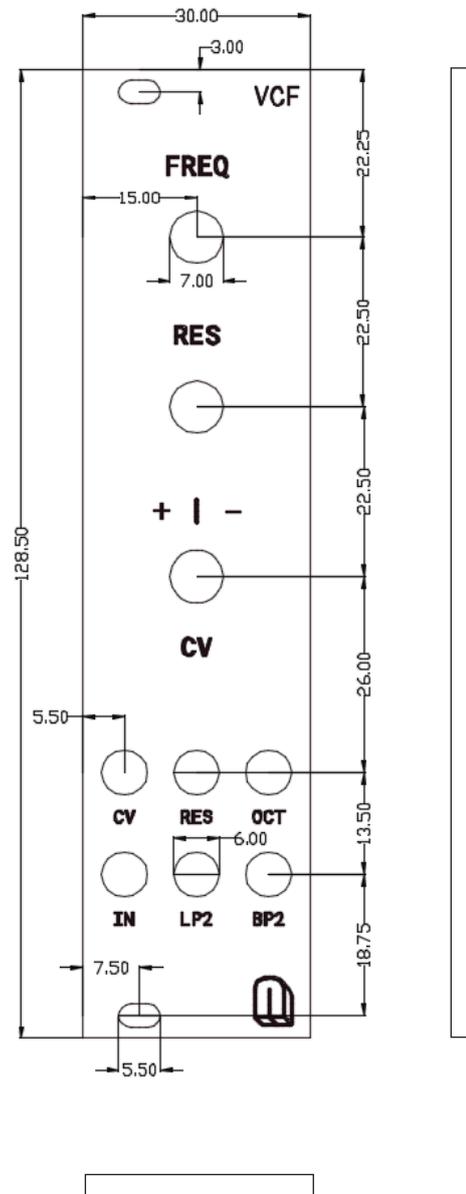
Escala				Nº de Hoja
1:1	Panel LFO. Alzado, planta y perfil.		Componentes Mecánicos.	3.2



1	Panel ADSR	128,5x30x3 mm	MDF 3mm	#003	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

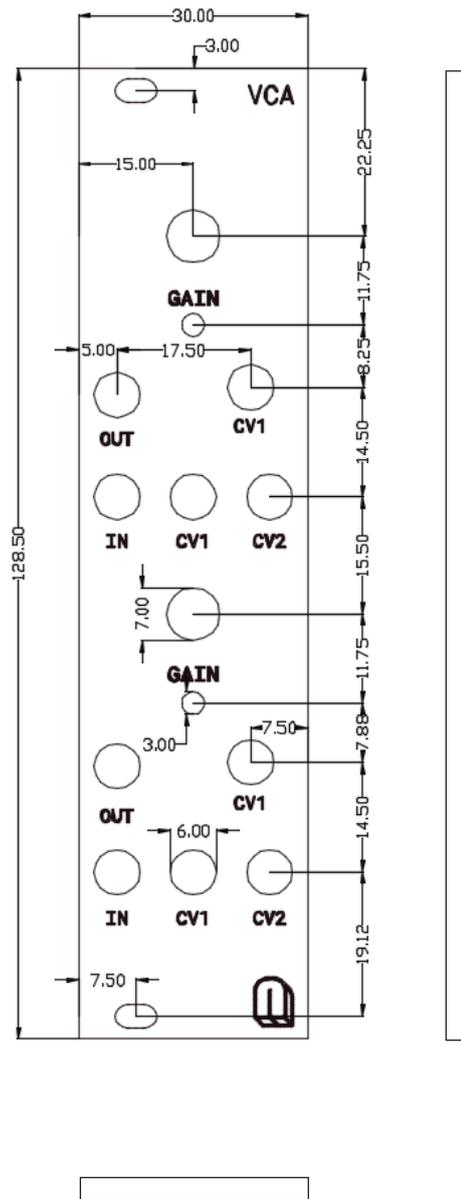
	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala	Panel ADSR. Alzado, planta y perfil.	Componentes Mecánicos.	Nº de Hoja
1:1			3.3



1	Panel VCF	128,5x30x3 mm	MDF 3mm	#001	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			
Escala	Panel VCF. Alzado, planta y perfil.			Componentes Mecánicos.	Nº de Hoja 3.4
1:1					



1	Panel VCA	128,5x30x3 mm	MDF 3mm	#002	1
Nº Pieza	Denominación	Dimensiones	Material	Referencia	Cantidad

	Fecha	Nombre	Firmas	TFG. Diseño e Implementación de un sintetizador de audio modular basado en síntesis substractiva.	 UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Diseñado por:	13/05/2018	Gutiérrez-Ravé			
Revisado por:	15/05/2018	Coll Arnau			

Escala		Componentes Mecánicos.	Nº de Hoja
1:1	Panel VCA. Alzado, planta y perfil.		3.5