



**Universidad  
de La Laguna**

**Escuela Superior  
de Ingeniería y Tecnología**

**Departamento de Ingeniería Industrial**

**ESCUELA SUPERIOR DE INGENIERÍA  
Y TECNOLOGÍA**

## **Trabajo de Fin de Grado**

**CETARRAY: Un array de hidrófonos inteligentes  
basado en una red de Raspberry Pi para la detección  
pasiva de cetáceos.**

**Titulación:** Grado en Ingeniería Electrónica Industrial y  
Automática

**Estudiante:** Óscar Pérez Díaz

**Tutor:** Fernando Luis Rosa González

1 de septiembre de 2018



**IMPRESO DE AUTORIZACIÓN DEL  
TRABAJO DE FIN DE GRADO POR EL  
TUTOR**

**Curso 2017/2018**

**D. Fernando Luis Rosa González, con D.N.I. 43611314-W, como tutor del estudiante D. Oscar Pérez Díaz en el Trabajo de Fin de Grado titulado CETARRAY: Un array de hidrófonos inteligentes basado en una red de Raspberry Pi para la detección pasiva de cetáceos.**

doy mi autorización, acreditada por la firma electrónica de este documento, para la presentación y defensa de dicho proyecto, a la vez que confirmo que la estudiante ha cumplido con los objetivos generales y particulares que lleva consigo la realización del mismo.

La Laguna, a 1 de Septiembre de 2018





**Universidad  
de La Laguna**

**Escuela Superior  
de Ingeniería y Tecnología**

**Departamento de Ingeniería Industrial**

**ESCUELA SUPERIOR DE INGENIERÍA  
Y TECNOLOGÍA**

## **Trabajo de Fin de Grado**

**CETARRAY: Un array de hidrófonos inteligentes  
basado en una red de Raspberry Pi para la detección  
pasiva de cetáceos.**

# **TOMO I Memoria**

**Titulación:** Grado en Ingeniería Electrónica Industrial y  
Automática

**Estudiante:** Óscar Pérez Díaz

**Tutor:** Fernando Luis Rosa González

1 de septiembre de 2018



# Índice general

<b>I Memoria</b>	<b>5</b>
<b>Resumen</b>	<b>13</b>
<i>Abstract</i>	<b>15</b>
<b>1. Introducción</b>	<b>17</b>
1.1. Marco general . . . . .	17
1.2. Objetivos . . . . .	18
<b>2. Material</b>	<b>19</b>
2.1. Raspberry Pi . . . . .	19
2.2. Tarjeta de audio Cirrus Logic . . . . .	19
2.3. <i>Software</i> . . . . .	20
2.3.1. Git . . . . .	20
2.3.2. Python . . . . .	20
2.3.3. LaTeX . . . . .	20
2.3.4. LTSpice . . . . .	21
2.3.5. kiCAD . . . . .	21
2.3.6. FreeCAD . . . . .	22
<b>3. Diseño</b>	<b>23</b>
3.1. Etapa de adquisición . . . . .	23
3.2. Etapa de procesamiento . . . . .	24
3.3. Etapa de adaptación . . . . .	28
3.3.1. Simulación y test del circuito . . . . .	31
3.3.2. Diseño de la placa de circuitos impresos . . . . .	33
3.4. Diseño mecánico . . . . .	36
<b>4. Fabricación</b>	<b>41</b>
4.1. Placa de circuitos impresos . . . . .	41
4.2. Elementos mecánicos . . . . .	46

<b>5. Resultados y conclusiones</b>	<b>49</b>
<b>6. <i>Results and conclusions</i></b>	<b>51</b>
<b>II Pliego de condiciones y presupuesto</b>	<b>55</b>
<b>7. Pliego de condiciones</b>	<b>57</b>
<b>8. Presupuesto</b>	<b>59</b>
<b>III Anexos</b>	<b>61</b>
<b>9. Planos</b>	<b>63</b>
9.1. Etapa de adaptación . . . . .	63
9.2. Placa de circuitos impresos . . . . .	66
9.3. Elementos mecánicos . . . . .	68
<b>10. <i>Datasheets</i></b>	<b>77</b>
10.1. <i>Cirrus Logic Audio Card</i> . . . . .	77
10.2. OPA2354 . . . . .	82
10.3. Hidrófono ITC-6050C . . . . .	84
10.4. Electret . . . . .	86

# Índice de figuras

3.1.	Esquema de funcionamiento del electret. . . . .	24
3.2.	Señal mono de 880 Hz muestreada a 16000 Hz. . . . .	26
3.3.	Transformada de fourier de la señal mono de 880 Hz muestreada a 16000 Hz. . . . .	26
3.4.	Análisis SNR de varios tonos a distintas frecuencias de muestreo. . . . .	27
3.5.	Estimación de posiciones teóricas (a la izquierda) y prueba de estimación de posiciones reales (a la derecha). . . . .	27
3.6.	Esquema electrónico del circuito implementado. . . . .	28
3.7.	Esquema electrónico del amplificador. . . . .	29
3.8.	Respuesta del filtro simulado en Python. . . . .	30
3.9.	Esquema electrónico del filtro. . . . .	30
3.10.	Entrada de la simulación del circuito en LTSpice. . . . .	31
3.11.	Salida de la simulación del circuito en LTSpice. . . . .	32
3.12.	Montaje del circuito en placa <i>protoboard</i> . . . . .	32
3.13.	Asignación de componentes con CvPcb. . . . .	33
3.14.	Distribución final de los componentes en PcbNew. . . . .	34
3.15.	Trazado de pistas y tabla de anchos empleados. . . . .	35
3.16.	Placa de circuitos completa. . . . .	35
3.17.	Integrados del OPA2354 y sus correspondientes pines. . . . .	37
3.18.	Edición de las características de un pin. . . . .	38
3.19.	Render de la caja de la Raspberry Pi y la tarjeta de audio. . . . .	39
3.20.	Render de la caja del amplificador. . . . .	39
3.21.	Render del diseño mecánico completo. . . . .	40
4.1.	Plancha para las placas de circuitos impresos. . . . .	42
4.2.	Revisión tras el planchado de una placa de circuitos impresos con la lupa. . . . .	43
4.3.	Tanque para el atacado químico y lavado de la placa. . . . .	44
4.4.	Horno para las placas de circuitos impresos. . . . .	44
4.5.	Operacional desoldado para su colocación. . . . .	45
4.6.	Preparado de la cama para la impresión de una pieza. . . . .	46

4.7. Algunas piezas impresas. . . . . 47

# Índice de tablas

8.1. Presupuesto. <b>Uni.:</b> Unidad, <b>€/ud:</b> Precio por unidad, <b>€:</b> Total en euros. . . . .	59
8.2. Precio detallado de los componentes electrónicos. <b>Uni.:</b> Unidad, <b>€/ud:</b> Precio por unidad, <b>€:</b> Total en euros. . . . .	60
8.3. Precio detallado de los elementos mecánicos. <b>Uni.:</b> Unidad, <b>€/ud:</b> Precio por unidad, <b>€:</b> Total en euros. . . . .	60



# Resumen

En el presente Trabajo de Fin de Grado se ha realizado el prototipado de un array de hidrófonos para la detección de cetáceos. En el diseño se incluye tanto el diseño electrónico, que incorpora la adquisición, adaptación y procesamiento de las señales, y el diseño de la mecánica. También se ha realizado la fabricación de un prototipo aéreo que permite demostrar el funcionamiento del sistema para la localización de fuentes.



## ***Abstract***

*In the present Final Degree Project it has been carried out the prototyping of an array of hydrophones for cetaceans detection. The design includes both the electronic design, which incorporates the acquisition, adaptation and signal processing, and the mechanical design. It has also been done the manufacture of the aerial prototype that let test the localization procedure to localize sources.*



# Capítulo 1

## Introducción

### 1.1. Marco general

La bioacústica ha mejorado nuestra comprensión de la evolución, la taxonomía, la conservación de la fauna y la fisiología animal [2]. Muchas aves e invertebrados producen canciones territoriales, los murciélagos (*Chiroptera Microchiroptera*) usan ultrasonidos para detectar presas y los elefantes (*Loxodonta*) usan infrasonidos para comunicarse. Las llamadas y canciones a menudo son características de una especie y, en muchos casos, transmiten las características biológicas, de comportamiento y ecológicas de la fuente. Las grabaciones acústicas pueden, por lo tanto, revelar una gran cantidad de información sobre individuos, poblaciones y el medio ambiente [10].

Los sistemas en chip de bajo coste se han vuelto ampliamente disponibles en la última década. Por ejemplo, la Raspberry Pi, que se desarrolló originalmente como una herramienta educativa, se ha adaptado para una amplia variedad de aplicaciones. Estos y otros dispositivos similares, como la placa de desarrollo BeagleBone Black, consumen potencia mínima y usan hardware de alta especificación en relación con su pequeño tamaño y bajo coste. Además, son operados utilizando sistemas operativos basados en Unix, de código abierto, libremente distribuidos y fácilmente disponibles y pueden ser alimentados por cualquier batería, como dispositivos de carga USB o baterías de vehículos. Estas características hacen que los sistemas en chip, como la Raspberry Pi, altamente adaptables, y tienen muchas aplicaciones potenciales en ecología [10].

La utilización de la detección acústica pasiva en el medio marino ha permitido conocer los hábitos de las colonias de cetáceos en la zona de Canarias. Para ello se han utilizado arrays lineales de hidrófonos que permiten adquirir el sonido y almacenarlo para su posterior estudio. La gran cantidad de datos, así como el enorme coste humano necesario para su revisión han hecho que desde hace algunos años

se haya empezado a automatizar cierto tipo de tareas como es la detección de actividad acústica diferenciada de la actividad sonora presente de modo regular. La elaboración de algoritmos eficientes de detección de vocalizaciones de cetáceos en el grupo del departamento, así como la instalación de una red de hidrófonos inteligentes en las piscinas de Loro Parque permiten avanzar el siguiente paso en el desarrollo que es la localización de las fuentes de emisión. Para ello se propone realizar un array de hidrófonos inteligentes utilizando tecnología de disposición pública de bajo coste como es la Raspberry Pi.

## 1.2. Objetivos

Los objetivos de la realización del presente proyecto se exponen a continuación:

- Diseño de un array, incluyendo tanto su parte electrónica como mecánica.
- Programación de dicho array para la detección de las fuentes sonoras.
- Obtención de un prototipo funcional de dicho array, empleando para ello los medios disponibles en el laboratorio.

# Capítulo 2

## Material

En este apartado se exponen los elementos que han servido como base para el diseño del array. Se incluyen tanto elementos de *hardware*, como la Raspberry Pi y la tarjeta de audio Cirrus Logic, que son el pilar del proyecto; como elementos de *software*, empleados para generar diseños, tanto mecánicos como electrónicos, y programas u otros archivos necesarios, como puede ser la memoria.

### 2.1. Raspberry Pi

La Raspberry Pi es un sistema en chip o *System on a Chip (SOC)*, es decir, que todos los módulos que componen el sistema están integrados en un mismo chip o circuito integrado. Existen diversas versiones del *hardware*, en este caso se empleará el modelo Raspberry Pi 3 Modelo B. En lo referente al *software* del sistema, se emplea cualquier sistema operativo, aunque lo más común es el uso de alguna versión de GNU/Linux.

Se ha elegido la Raspberry debido a que presenta mayores prestaciones que otros microcontroladores, además de tener mayor versatilidad, ya que permite programar en múltiples lenguajes de programación y añadir accesorios o módulos externos para aumentar sus funcionalidades, como puede ser una tarjeta de audio como la empleada en el proyecto.

### 2.2. Tarjeta de audio Cirrus Logic

Se trata de una tarjeta de audio desarrollada por Wolfson Microelectronics para varios modelos de Raspberry Pi, entre los que se incluye el modelo B. Cuenta con múltiples entradas y salidas de audio tanto analógicas como digitales (SPDIF) y micrófonos DMIC integrados, entre otras funciones propias de una tarjeta de audio.

Ya que la Raspberry Pi no incorpora entradas de audio se ha añadido esta tarjeta, que permite la entrada de audio de alta resolución, para posibilitar la realización del proyecto.

## 2.3. Software

El software empleado ha sido principalmente *software* libre. Este *software*, por elección expresa de su autor, puede ser ejecutado, copiado, estudiado, mejorado o redistribuido, por tanto, el acceso a su código es libre y, normalmente, gratuito.

### 2.3.1. Git

Git es un sistema de control de versiones gratuito y de *software* libre. Este sistema permite crear un repositorio donde se depositan los archivos y se registran y coordinan los cambios realizados por cualquiera de los participantes en el proyecto.

A lo largo de la realización del TFG se ha empleado un repositorio, ubicado en un servidor remoto, para facilitar la supervisión del trabajo por parte del tutor, así como tener un registro de los cambios realizados, ordenados y guardados.

### 2.3.2. Python

Python es un lenguaje de programación interpretado de alto nivel. "Se desarrolla bajo una licencia de código abierto aprobada por OSI, por lo que es libremente utilizable y distribuible, incluso para uso comercial [9]".

Este lenguaje de programación contiene multitud de extensiones, durante la realización del proyecto se han empleado las siguientes: NumPy, que ofrece mayor soporte matemático a las funciones de Python; SciPy, librería fundamental para trabajar con señales y filtros, entre otros; Matplotlib, para el trazo de gráficas; Soundfile, librería de audio.

Normalmente, para facilitar la programación, se ha empleado el modo interactivo de Python: iPython.

### 2.3.3. LaTeX

*LaTeX* es un sistema de preparación de documentos centrado en el contenido del texto. Al contrario que otros procesadores de texto como *Microsoft Word* o *LibreOffice Writer*, donde, mientras se escribe, es posible ver el resultado final en *LaTeX* se escribe un texto plano en el que se emplean diversas macros para dar

formato al mismo. Esto permite al autor centrarse en el contenido del texto sin tener que preocuparse por el formato.

Para editar los textos se ha utilizado la aplicación *Kile*, que proporciona un entorno de edición *LaTeX* amigable, desarrollada por KDE, comunidad internacional que desarrolla *software* libre.

### 2.3.4. LTSpice

LTSpice es un software de simulación basado en SPICE que permite la edición de esquemáticos y el análisis de señales de circuitos. Además, incluye gran cantidad de componentes de Analog Devices y librerías de componentes reales para la simulación de los circuitos [6].

### 2.3.5. kiCAD

KiCad es un paquete de *software* de código abierto para *Electronic Design Automation* (EDA). Los programas manejan la captura de esquemáticos y el diseño de PCB con salida de Gerber [5].

Este entorno de software libre permite la creación y edición de esquemas eléctricos para el diseño de placas de circuitos impresos o *Printed Circuit Board* (PCB). Incluye múltiples herramientas para cumplir con estas funcionalidades.

#### Eeschema

Esta herramienta permite el diseño de esquemas eléctricos y la creación de distintos ficheros: *Netlist*, que indica las conexiones entre los componentes del circuito; *Bill of Materials*, una lista en la que se indican los valores de los componentes, su referencia y la cantidad; y *Cross References*, muestra una lista de los componentes con su nombre, referencia, valor, librería y sus coordenadas en el esquemático. Además, posee una herramienta para la comprobación de las reglas eléctricas o *Electrical Rules Check (ERC)*, que verifica conflictos entre pines o pines no conectados.

También es necesario destacar que las librerías, de símbolos, huellas y modelos 3D, de KiCAD mejoran constantemente debido a que son mantenidas por la propia comunidad [5].

#### CvPcb

Con la herramienta CvPcb se asocian los componentes utilizados en el esquema eléctrico con sus huellas o *footprints*, necesarios para elaborar el diseño de

la placa de circuito impreso. De este modo se edita el Netlist, que contendrá las huellas asociadas a cada componente.

### **Pcbnew**

Pcbnew es la herramienta que permite la creación y edición de placas de circuito impreso en KiCAD. Permite trabajar sobre 32 capas de cobre, 14 capas técnicas (bordes, máscara de soldadura, pasta de soldadura, serigrafía) y 4 capas auxiliares.

Después de generar el esquema eléctrico, asociar los componentes y generar el Netlist. Este último se carga en Pcbnew, obteniendo las huellas de los componentes y sus conexiones para el diseño de la placa de circuito impreso. Para facilitar el diseño, se muestran las conexiones entre los componentes o *rats nets*.

Finalmente, permite obtener los ficheros necesarios para la fabricación de la placa mediante diferentes métodos: ficheros gerber (.gbl, .gtl, .gtp, .gto, .gts, .gtr), ficheros de taladro (.drl), o imágenes de las capas (.svg)

### **2.3.6. FreeCAD**

FreeCAD es un modelador 3D paramétrico de propósito general. El desarrollo es completamente de código abierto (licencia LGPL). FreeCAD está dirigido directamente a la ingeniería mecánica y al diseño de productos, pero también se adapta a una gama más amplia de usos en torno a la ingeniería, como la arquitectura u otras especialidades de ingeniería.[...] FreeCAD hace un uso intensivo de todas las grandes bibliotecas de código abierto que existen en el campo de la Informática Científica. Entre ellos se encuentran OpenCascade, un potente kernel de CAD, Coin3D, una encarnación de Open Inventor, Qt, el marco de interfaz de usuario de fama mundial y Python, uno de los mejores lenguajes de scripting disponibles. FreeCAD también puede ser usado como biblioteca por otros programas [1].

El programa permite el diseño de elementos mecánicos empleando múltiples métodos como: creación de figuras geométricas simples y operaciones entre estas, creación de croquis en 2D y su posterior transformación en 3D, entre otros. Todo esto pudiendo cambiar la forma de trabajo, adecuándola a la de otros software de diseño.

# Capítulo 3

## Diseño

En este capítulo se incluyen los procesos de diseño y la generación de diversos elementos necesarios para la obtención del array. El diseño se divide en dos partes principales. La primera, referente a la obtención y gestión de datos, se encuentra subdividida en tres partes: una etapa de adquisición, necesaria para detectar los sonidos y conformada por micrófonos electret; una etapa de procesamiento, en la que se emplea el conjunto formado por la Raspberry Pi y la tarjeta de audio Cirrus Logic para gestionar y analizar la información adquirida; y una etapa de adaptación, que permite adecuar las señales entre las dos etapas anteriores gracias a un circuito electrónico, para el cual se ha diseñado una placa de circuitos impresos. La segunda parte incluye el diseño mecánico, incorpora el diseño de la estructura del array, así como sus características principales y de las partes que la conforman.

### 3.1. Etapa de adquisición

La etapa de adquisición, tal y como su nombre indica, permite la adquisición de señales de audio que procesar posteriormente. Para ello, es necesario un elemento capaz de detectar las ondas sonoras y transformarlas en una señal eléctrica, ya sea analógica o digital.

Para el diseño original se pretendía emplear un hidrófono modelo ITC-6050C, que tiene una etapa de preamplificación que adecúa la señal analógica a niveles nominales de audio, una buena respuesta en frecuencia y como su nombre indica es sumergible, estas características se pueden observar en la sección 10.3. Sin embargo, debido a su escasa disponibilidad se ha prototipado el sistema con micrófonos electret.

Los micrófonos electret son baratos, sencillos de utilizar y su respuesta en frecuencia es bastante aceptable, a diferencia de la de otros micrófonos de con-

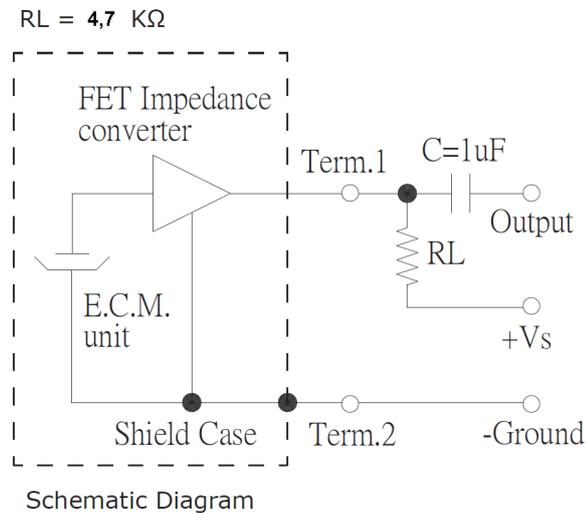


Figura 3.1: Esquema de funcionamiento del electret.

densador. Al contrario que el hidrófono, la señal de audio que devuelve el electret es muy débil, de modo que será necesaria una etapa de adaptación.

Para asegurar el correcto funcionamiento del electret, del que no se conocía el modelo exacto, se realizaron varias medidas de laboratorio para calcular tanto su consumo como su resistencia interna, realizando el montaje de la figura 3.1.

De este modo, se ha determinado que la resistencia interna del electret es de 7,82 kohm y su consumo de 0,4 mA, operando a 3,124 V. Para simplificar el diseño, esta configuración es la que se empleará en el array, junto con la correspondiente etapa de adaptación explicada en este capítulo.

## 3.2. Etapa de procesamiento

La etapa de procesamiento, tras ser debidamente configurada, se encarga de controlar la grabación, guardar la información recibida y analizar la señal recibida gracias a múltiples scripts, funciones del propio sistema y programas desarrollados en Python.

Previamente a trabajar con la Raspberry Pi esta se ha configurado para poder trabajar en ella sin la necesidad de incorporar más periféricos de los necesarios, ya que podrían afectar a las medidas y limitar la funcionalidad del array, utilizado para ello una conexión vía SSH (*Secure SHell*). Un protocolo SSH permite acceder al sistema y manejarlo mediante una ventana de comandos. Además, también permite redirigir el tráfico del Sistema de Ventanas X, que se encarga de gestionar las ventanas en sistemas operativos UNIX, permitiendo ejecutar programas

gráficos [7]. Para ello, se ha fijado la dirección IP del Wi-Fi de la Raspberry en el router del laboratorio y se ha configurado la Raspberry [8] para poder usar el SSH. También es necesario seleccionar la entrada de señal de la tarjeta de audio que se va a emplear, en este caso el jack de 3.5mm. Para ello, la Cirrus Logic incorpora una serie de scripts que permiten activar las diferentes entradas o salidas de la tarjeta de audio. En este caso, el puerto de entrada es el LineIn, así que en la terminal de la Raspberry Pi ejecutamos el comando indicado en la hoja de datos de la sección 10.1. Una vez seleccionada la entrada que se quiere utilizar, se graba con el comando **arecord**.

El comando **arecord** permite controlar diversos aspectos de la grabación: formato de muestreo, en este caso se ha empleado un *little endian* de 16 bit; canales de grabación, este valor puede estar entre 1 y 32; frecuencia de muestreo, normalmente se empleará 192kHz; o tiempo de grabación, indicando un tiempo en segundos la grabación queda interrumpida automáticamente. Los archivos de audio quedan guardados en el directorio en el que hayamos ejecutado el comando **arecord** con el nombre que hayamos indicado al ejecutarlo.

Tras configurar la Raspberry Pi y adquirir los archivos de audio, es necesario extraer información útil de la grabación. Por ello, se han desarrollado una serie de programas en python que permiten analizar la señal y extraer sus características; además, se ha desarrollado otro programa con el que estimar la posición del origen del sonido.

El programa para el análisis de la señal contiene varias funciones que permiten ver los dos canales de audio que se han empleado en la grabación (figura 3.2) y su espectro de fourier de la señal (figura 3.3). Además, incluye otra función que analiza la relación señal-ruido o *Signal Noise Ratio* (SNR) y puede devolver una gráfica comparativa entre muestreos a distintas frecuencias y diferentes señales 3.4.

Finalmente, se ha desarrollado un programa para la estimación de la posición basado en la diferencia de tiempo en la que la señal es adquirida por cada micrófono. Teniendo esta información y conociendo la velocidad de propagación del sonido en el medio de trabajo podemos determinar la diferencia de distancia. De este modo es posible plantear una ecuación hiperbólica, haciendo la diferencia de distancias entre el punto de origen y cada micrófono. Así, con dicha diferencia es posible obtener una serie de puntos, que dan lugar a una hipérbola, donde está la posible posición del origen. Este programa ha sido empleado en la calibración del array para la detección de sonidos, como podemos comprobar en la figura 3.5. Con el diseño actual sólo es posible estimar la posición, sin embargo, añadiendo un tercer micrófono y calculando el punto de corte de las hipérbolas se puede determinar la posición exacta del punto de origen en el plano.

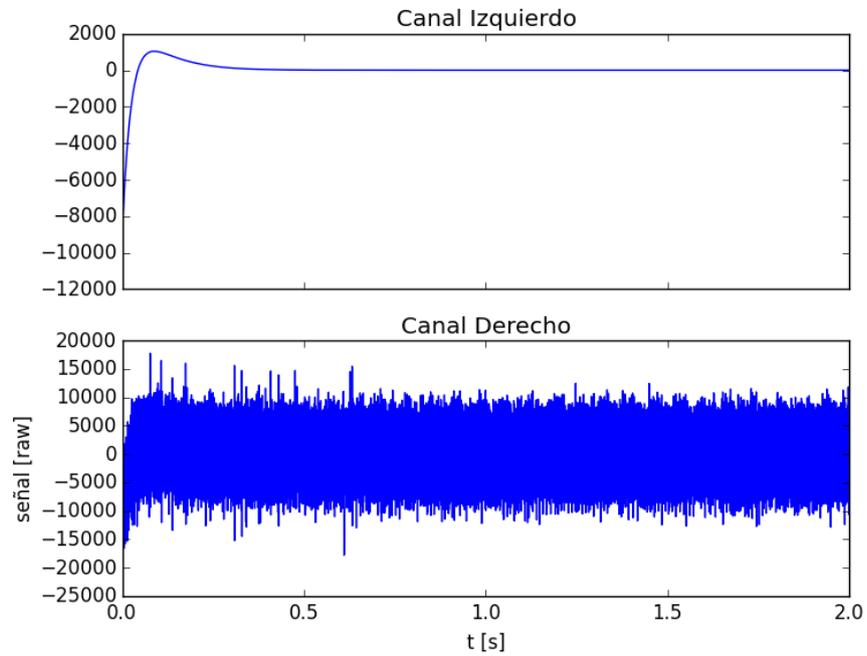


Figura 3.2: Señal mono de 880 Hz muestreada a 16000 Hz.

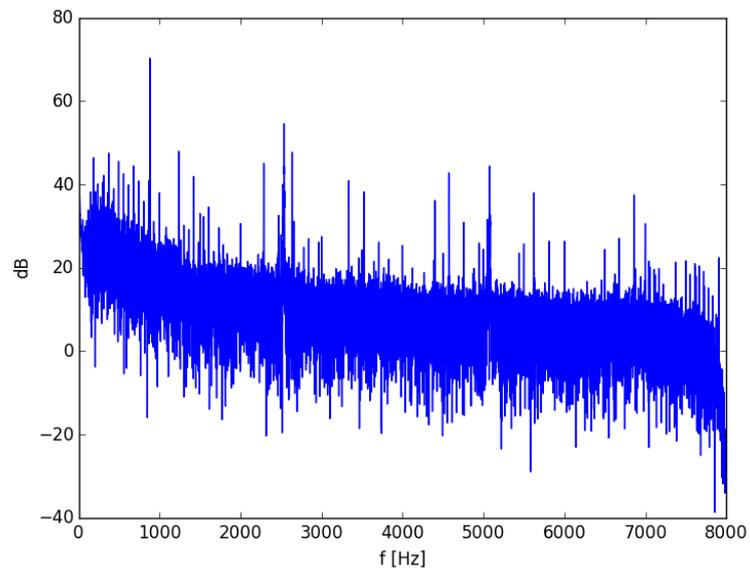


Figura 3.3: Transformada de fourier de la señal mono de 880 Hz muestreada a 16000 Hz.

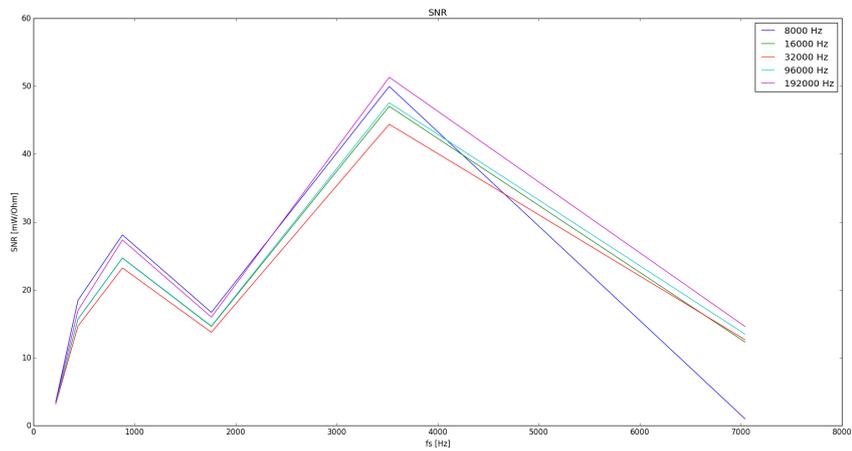


Figura 3.4: Análisis SNR de varios tonos a distintas frecuencias de muestreo.

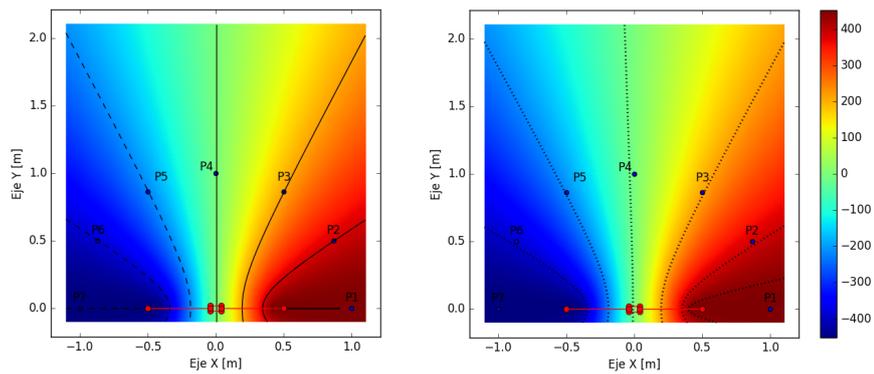


Figura 3.5: Estimación de posiciones teóricas (a la izquierda) y prueba de estimación de posiciones reales (a la derecha).

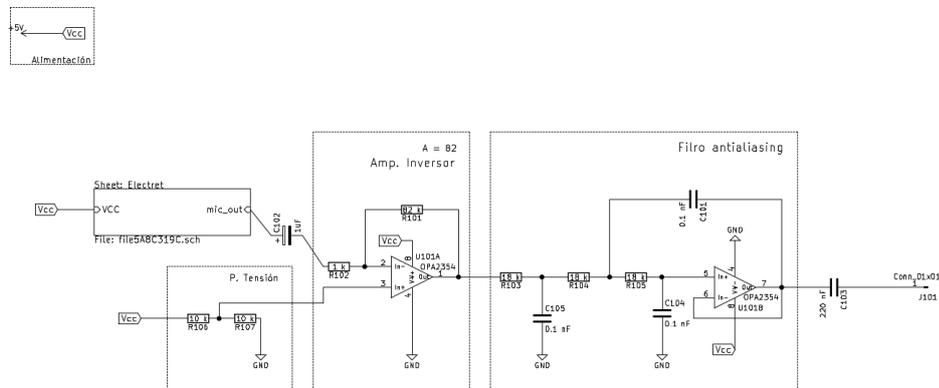


Figura 3.6: Esquema electrónico del circuito implementado.

### 3.3. Etapa de adaptación

Es necesario adaptar la salida de la etapa de adquisición a la etapa de procesamiento ya que cada una de ellas trabaja en amplitudes de señal muy diferentes. Para una correcta adaptación de la señal del electret es necesario amplificarla y filtrarla para eliminar los efectos del aliasing. Para ello, se ha diseñado, simulado y montado el circuito electrónico de la figura 3.6 basado en amplificadores operacionales con los que adecuar la señal proveniente de los electret a la entrada analógica de la tarjeta de audio.

Para el diseño de las partes que configuran esta etapa se ha tenido en cuenta que la tensión de alimentación es una de las tensiones que puede proporcionar la raspberry para no tener que utilizar reguladores de tensión u otros elementos adicionales. De este modo, nuestra tensión de alimentación será de 5 Vdc. Además, la Raspberry Pi no proporciona tensiones negativas, por tanto los operacionales quedan alimentados entre tierra y 5 Vdc.

Como se ha comprobado, de la etapa de adquisición se obtiene una señal con una amplitud de 12,6 milivoltios. En la documentación de la tarjeta de audio Cirrus Logic es posible comprobar que el nivel máximo de entrada en el puerto Line In es de 1 Vrms (ver datasheet en la sección 10.1). Por tanto, en la etapa de adaptación se debe aumentar la amplitud de la señal en una relación de 83,3 veces.

En consecuencia, se ha decidido incorporar un amplificador operacional en configuración inversora con una ganancia de 82, obteniendo a su salida una señal con 1 V de amplitud. Teniendo en cuenta que la alimentación del operacional es unipolar, en su entrada positiva se coloca un partidor de tensión, ubicando el nivel de continua en 2,5 Vdc, evitando que la señal esté en la zona de saturación del operacional.

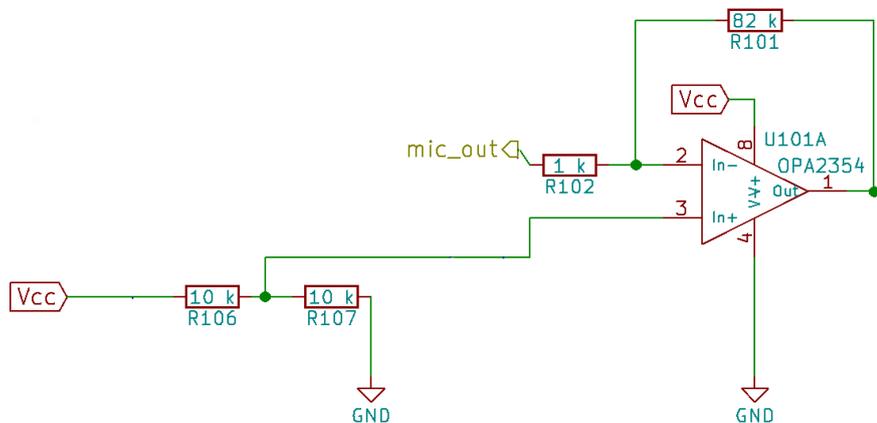


Figura 3.7: Esquema electrónico del amplificador.

El muestreo de la señal se realiza a 192 KHz, por tanto, el espectro de la señal se repetirá cada  $N \cdot 192 \text{ kHz}$  (siendo  $N$  un número entero) en el espectro de frecuencia; y la frecuencia de Nyquist será de 96 kHz. Teniendo todo esto en cuenta, nuestro filtro ha de tener la frecuencia de corte en 96 kHz para evitar que se produzca aliasing.

Para el diseño del filtro antialiasing se ha utilizado un filtro de Butterworth paso bajo de tercer orden con ganancia 1, es decir, configurado como seguidor de tensión.

Con las especificaciones indicadas podemos estimar un comportamiento del filtro con Python utilizando la librería SciPy, el resultado se puede observar en la figura 3.8.

Se ha decidido utilizar componentes equilibrados, de este modo el filtro presentará un polo triple en la frecuencia de corte elegida [4].

$$f = \frac{1}{2\pi \cdot R \cdot C} \quad (3.1)$$

Sustituyendo en la fórmula anterior y teniendo en cuenta valores de componentes comerciales obtenemos una frecuencia de corte de 88419,41 Hz con el valor de la resistencia en 18 kohm y el valor del capacitor en 100 pF.

De este modo obtenemos un circuito que cubre todas las necesidades de adaptación de la señal.

El elemento más importante y limitador del diseño es el amplificador operacional. Para poder fabricar el circuito y que este sea funcional, el amplificador operacional ha de cumplir diversos requerimientos: el producto ganancia-ancho

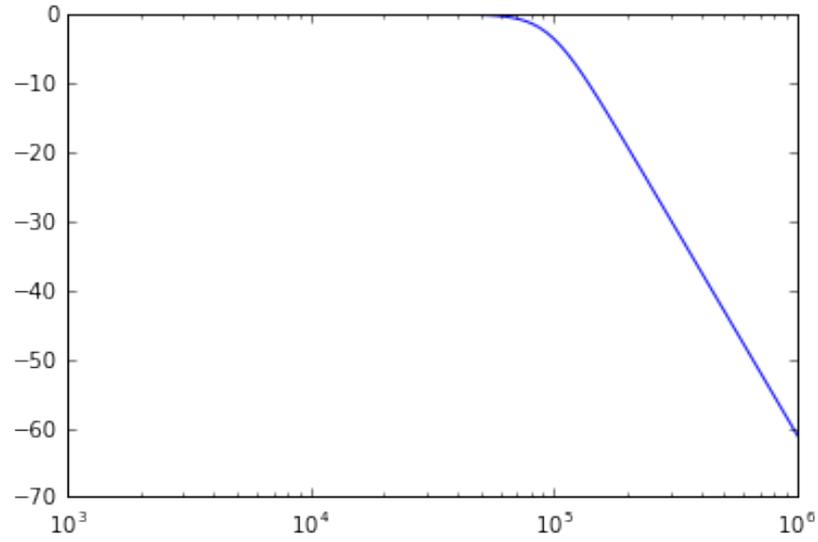


Figura 3.8: Respuesta del filtro simulado en Python.

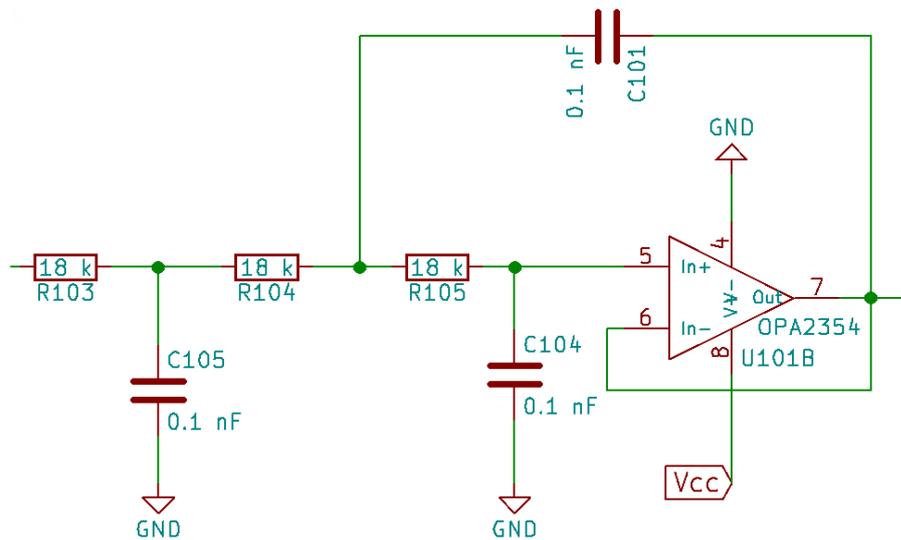


Figura 3.9: Esquema electrónico del filtro.

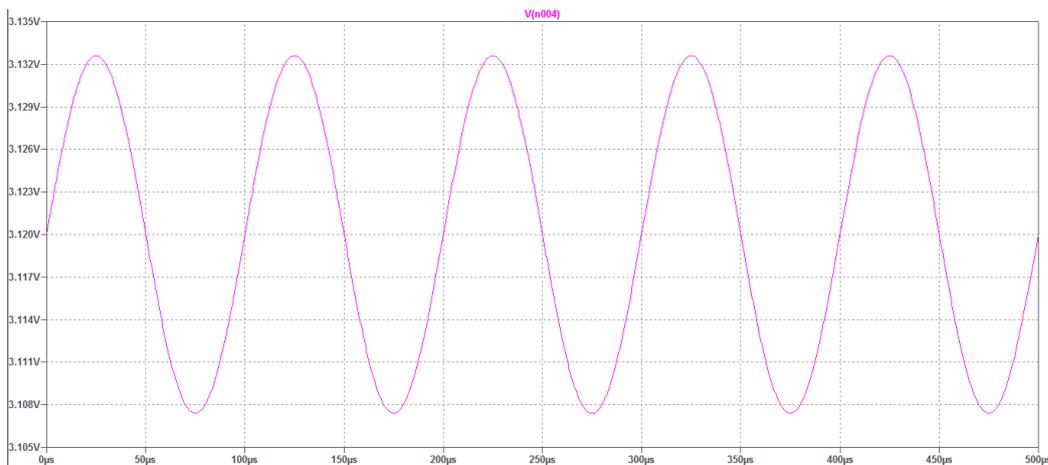


Figura 3.10: Entrada de la simulación del circuito en LTSpice.

de banda ha de ser muy grande, ya que se necesita un ancho de banda de 96 kHz y la ganancia del circuito está en torno a las centenas; debe ser unipolar con un rango de alimentación bajo, puesto que estará alimentado entre tierra y 5 Vdc; y presentar ruido bajo, para trabajar con señales de audio. Es por ello, que se ha elegido un OPA2354 (ver 10.2), que cumple con todos los requerimientos del circuito, y además tiene dos amplificadores en su encapsulado.

### 3.3.1. Simulación y test del circuito

Para asegurar el correcto funcionamiento del circuito antes de la fabricación la placa de circuito impreso se ha llevado a cabo una simulación del diseño y también se ha montado con componentes reales en una placa *protoboard* para observar el comportamiento del circuito. En ambos casos se ha utilizado una fuente de tensión y un condensador para sustituir el conjunto del electret. Además, se ha comprobado también la respuesta del electret para garantizar una respuesta del conjunto adecuada.

En cuanto a la simulación del circuito, realizada en LTSpice, los resultados obtenidos han sido los esperados. Se puede comprobar en la figura 3.10 una supuesta señal generada por el Electret, con una amplitud de 12,6 mV y un valor de continua de 3,12 Vdc; y también es apreciable en la figura 3.11 que los valores de la salida del circuito están dentro de los límites elegidos.

Para el montaje en la *protoboard*, observable en la figura 3.12, se ha empleado otro operacional con un comportamiento similar al OPA2354 pero con unas prestaciones menores, por lo que fue necesario aumentar la tensión de alimentación para que pudiese funcionar. Aún así, se obtuvieron los resultados esperados.

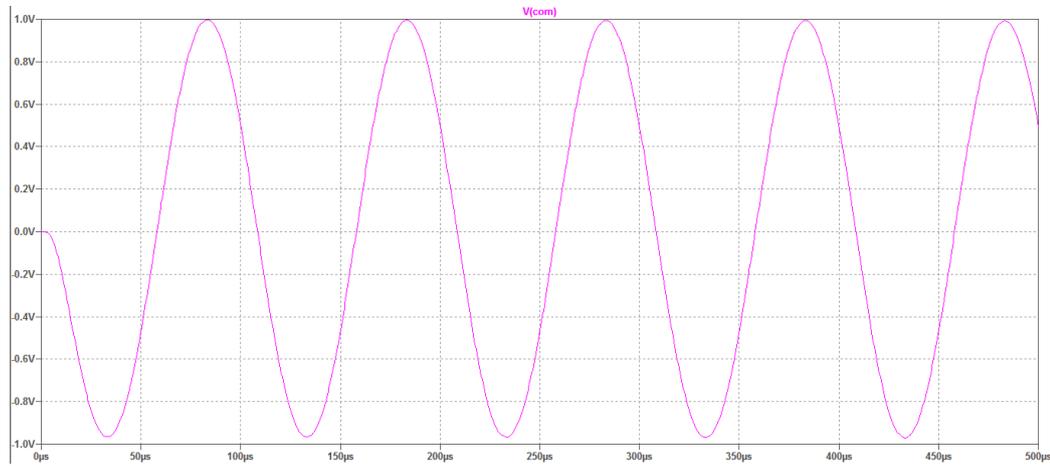


Figura 3.11: Salida de la simulación del circuito en LTSpice.

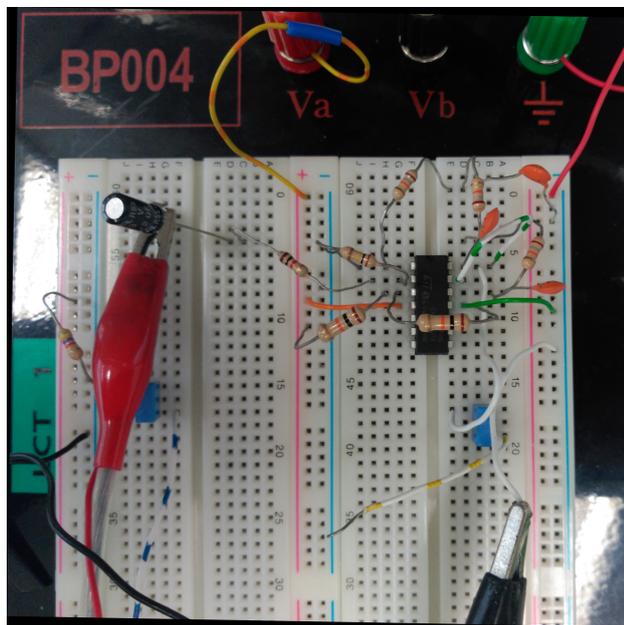


Figura 3.12: Montaje del circuito en placa *proto*board.

Component	Value	Footprint	Mounting
C101	0.1 nF	Capacitors_THT:C_Disc_D7.5mm_W5.0mm_P5.00mm	
C102	1uF	Capacitors_Tantalum_SMD:CP_Tantalum_Case-A_EIA-3216-18_Hand	
C103	220 nF	Capacitors_SMD:C_0805_HandSoldering	
C104	0.1 nF	Capacitors_THT:C_Disc_D7.5mm_W5.0mm_P5.00mm	
C105	0.1 nF	Capacitors_THT:C_Disc_D7.5mm_W5.0mm_P5.00mm	
J101	Conn_01x01_Male	Pin_Headers:Pin_Header_Angled_1x01_Pitch2.00mm	
MK201	Microphone Condenser	Buttons_Switches_SMD:SW_SPST_83S-1000	
Q201	Q_NFET_GDS	Buttons_Switches_THT:SW_E-Switch_EG1224_SPDT_Angled	
R101	82 k	Resistors_SMD:R_1210_HandSoldering	
R102	1 k	Resistors_SMD:R_1210_HandSoldering	
R103	18 k	Resistors_SMD:R_1210_HandSoldering	
R104	18 k	Resistors_SMD:R_1210_HandSoldering	
R105	18 k	Resistors_SMD:R_1210_HandSoldering	
R106	10 k	Resistors_SMD:R_0805_HandSoldering	
R107	10 k	Resistors_SMD:R_0805_HandSoldering	
R201	4.7 k	Resistors_SMD:R_1210_HandSoldering	
U101	OPA2354	Housings_SOIC:SOIC-8_3.9x4.9mm_Pitch1.27mm	
C0201		Capacitors_SMD:C_0201	
C0201		Capacitors_SMD:C_0201_NoSilk	
C0402		Capacitors_SMD:C_0402	
C0402		Capacitors_SMD:C_0402_NoSilk	
C0603		Capacitors_SMD:C_0603	
C0603		Capacitors_SMD:C_0603_HandSoldering	
C0805		Capacitors_SMD:C_0805	
C0805		Capacitors_SMD:C_0805_HandSoldering	
C1206		Capacitors_SMD:C_1206	
C1206		Capacitors_SMD:C_1206_HandSoldering	
C1210		Capacitors_SMD:C_1210	
C1210		Capacitors_SMD:C_1210_HandSoldering	
C1812		Capacitors_SMD:C_1812	
C1812		Capacitors_SMD:C_1812_HandSoldering	
C1825		Capacitors_SMD:C_1825	
C1825		Capacitors_SMD:C_1825_HandSoldering	
C2220		Capacitors_SMD:C_2220	
C2220		Capacitors_SMD:C_2220_HandSoldering	
C2225		Capacitors_SMD:C_2225	
C2225		Capacitors_SMD:C_2225_HandSoldering	
T284-A		Capacitors_SMD:C_Trimmer_Murata_T284-A	
T284-B		Capacitors_SMD:C_Trimmer_Murata_T284-B	
T2C3		Capacitors_SMD:C_Trimmer_Murata_T2C3	
T2R1		Capacitors_SMD:C_Trimmer_Murata_T2R1	
T2R4		Capacitors_SMD:C_Trimmer_Murata_T2R4	
T2Y2		Capacitors_SMD:C_Trimmer_Murata_T2Y2	
T2Y2		Capacitors_SMD:C_Trimmer_Sprague-Goodman_56G3	
T3N		Capacitors_SMD:C_Trimmer_Voltronics_T3N	
T30		Capacitors_SMD:C_Trimmer_Voltronics_T30	
T3R		Capacitors_SMD:C_Trimmer_Voltronics_T3R	
T3V		Capacitors_SMD:C_Trimmer_Voltronics_T3V	
T3Z		Capacitors_SMD:C_Trimmer_Voltronics_T3Z	
P7.50mm		Capacitors_THT:C_Axial_L3.8mm_D2.6mm_P7.50mm_Horizontal	
P10.00mm		Capacitors_THT:C_Axial_L3.8mm_D2.6mm_P10.00mm_Horizontal	
P12.50mm		Capacitors_THT:C_Axial_L3.8mm_D2.6mm_P12.50mm_Horizontal	
P15.00mm		Capacitors_THT:C_Axial_L3.8mm_D2.6mm_P15.00mm_Horizontal	
P15.00mm		Capacitors_THT:C_Axial_L5.1mm_D3.1mm_P7.50mm_Horizontal	
P15.00mm		Capacitors_THT:C_Axial_L5.1mm_D3.1mm_P10.00mm_Horizontal	
P15.00mm		Capacitors_THT:C_Axial_L5.1mm_D3.1mm_P12.50mm_Horizontal	
P15.00mm		Capacitors_THT:C_Axial_L5.1mm_D3.1mm_P15.00mm_Horizontal	

Figura 3.13: Asignación de componentes con CvPcb.

### 3.3.2. Diseño de la placa de circuitos impresos

Una vez diseñado y comprobado el circuito se ha procedido al diseño de una placa de circuitos impresos o *Printed Circuit Board* (PCB) utilizando para ello el programa de software libre kiCAD. La placa se ha diseñado teniendo en cuenta una serie de normas de diseño. Además, ha sido necesario crear una librería de componentes, ya que algunos de los empleados no estaban incluidos en las librerías de kiCAD.

El primer paso en el diseño de la placa es incluir el diseño electrónico del circuito en la parte correspondiente del kiCAD, Eeschema, indicando los valores de los componentes y asegurando que las conexiones son correctas para posteriormente, con la herramienta CvPcb, asociar los componentes del esquemático con los componentes reales con los que se fabricará la PCB. La asociación se muestra en la figura 3.13.

Tras completar el diseño del esquemático y asociar los componentes se exporta la *netlist* para poder cargarla desde la herramienta PcbNew y así mantener las conexiones virtuales entre los componentes.

Para el diseño de la placa no hay ninguna restricción, de tamaño, disipación o cualquier otra característica. Por tanto, se ha realizado tomando en cuenta las consideraciones del diseñador y siguiendo las reglas de diseño para la separación de pistas y componentes, y el ancho de pistas.

Una vez hecho todo esto, se pasa a la herramienta Pcbnew para el diseño de la placa de circuitos impresos. Aquí, importamos la *netlist* que se había generado

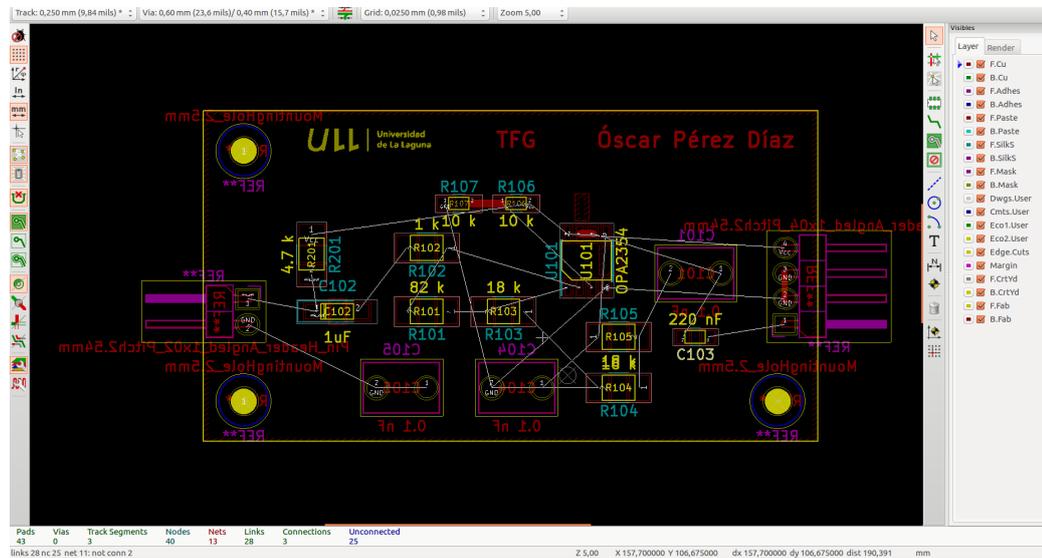


Figura 3.14: Distribución final de los componentes en PcbNew.

con la herramienta Eeschema, de este modo se cargan todos los componentes y sus conexiones correspondientes, que se muestran en el programa para facilitar la distribución de los mismos.

Normalmente, el primer paso en el diseño de la placa sería dibujar los bordes de acuerdo a las especificaciones. Sin embargo, como no hay ninguna restricción de dimensiones, se han colocado primero los componentes. Éstos se distribuyen de la forma más óptima posible siguiendo siempre las reglas de diseño. Además, se han añadido otros elementos como pines, agujeros de montaje, texto y el logo de la universidad (figura 3.14).

Una vez colocados los componentes, se procede a realizar el trazado de pistas. Para evitar efectos de diafonía o *crosstalk* se han trazado pistas relativamente anchas, de al menos 0,5 mm (ver figura 3.15; además, las pistas de alimentación tienen el ancho mínimo especificado por las reglas de diseño (1 mm).

Después de finalizar el trazado de pistas se traza un plano de tierra, que ocupará siempre el mayor área posible. Para ello, se emplea la herramienta de áreas del kiCAD, que permite controlar características como la separación entre el área y las pistas o los pads, en este caso de 0,254 mm, o el uso de nodos térmicos, empleados en el diseño.

De este modo la placa quedaría finalizada (figura 3.16), el último paso necesario es generar los archivos necesarios para la fabricación. En este caso, teniendo en cuenta la tecnología de fabricación empleada, se ha exportado el fotolito de la placa en formato SVG para su posterior impresión.

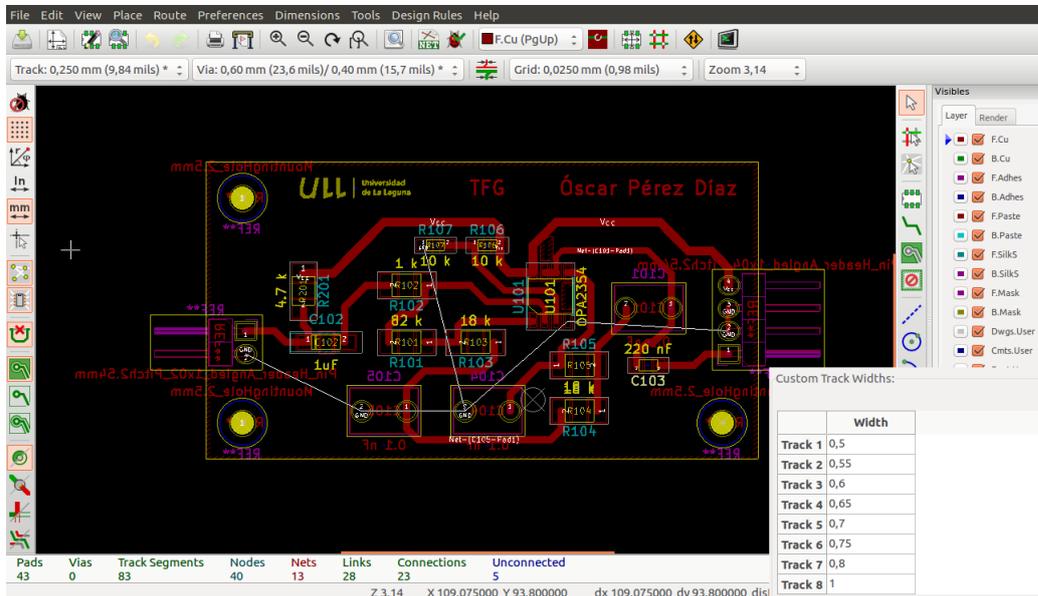


Figura 3.15: Trazado de pistas y tabla de anchos empleados.

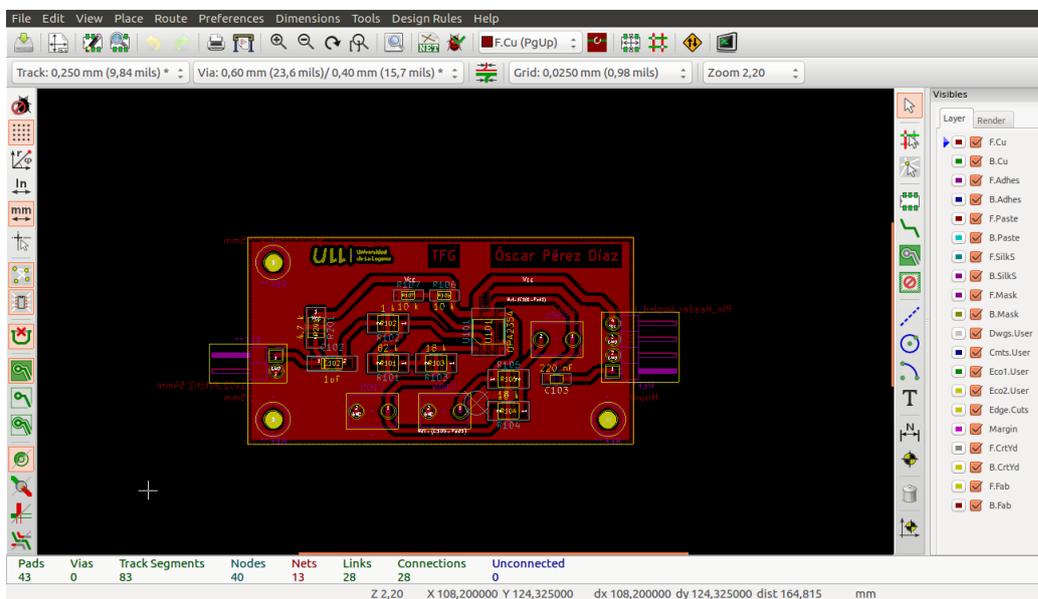


Figura 3.16: Placa de circuitos completa.

### Creación de un componente en kiCAD

KiCAD permite la creación y modificación de componentes y librerías, permitiendo añadir o cambiar tanto la huella como las conexiones electrónicas de los componentes que sea necesario. En este caso, como ya se ha indicado, fue necesario crear un componente nuevo para poder incluir el OPA2354 en el diseño electrónico.

La creación de un componente es un proceso simple, detallado en la documentación de KiCAD [3]. Una vez seleccionada la librería en la que queremos guardar el componente, creamos un nuevo componente utilizando el editor de librerías de kiCAD, que permite incluir varias partes dentro de un encapsulado, dibujar el cuerpo del componente y generar los distintos pines que deseamos, así como sus características 3.18, obteniendo finalmente el componente que queremos (figura 3.17).

## 3.4. Diseño mecánico

En este apartado se incluyen las especificaciones y requerimientos para el diseño de la mecánica del array, también incluye los elementos diseñados y una breve explicación sobre el procedimiento de diseño con la herramienta kiCAD.

El diseño mecánico del array supone múltiples retos, como la posibilidad de hacerlo sumergible o tener un array lo suficientemente grande para poder diferenciar mayor cantidad de posiciones.

Para resolver el problema de la sumergibilidad se han diseñado los diferentes elementos del array para poder incluir juntas tóricas y se ha planteado incluir elementos de sellado, como imprimaciones y silicona, a la hora de la fabricación.

Teniendo en cuenta las dimensiones del array, la velocidad del sonido en el medio en el que se trabaja y la frecuencia de muestreo se puede calcular la resolución que tendrá el array a la hora de estimar las posiciones. En primera instancia se ha planteado un array de una longitud de 1 m que se probará en el aire, por lo que la velocidad del sonido será de 340 m/s, muestreando a una frecuencia de 192kHz. Con estas características se obtiene una resolución de aproximadamente 560 muestras, es decir, es posible diferenciar 560 posiciones diferentes.

Ya que el array se probará en la superficie se ha decidido incluir varios elementos para facilitar su uso, como un asa, y aumentar la rigidez, añadiendo varios elementos de soporte.

Para la realización del diseño y los planos se ha empleado el programa FreeCAD, una vez definidas las características del montaje se han diseñado las partes necesarias para la fabricación del array y se ha realizado un diseño del montaje final.

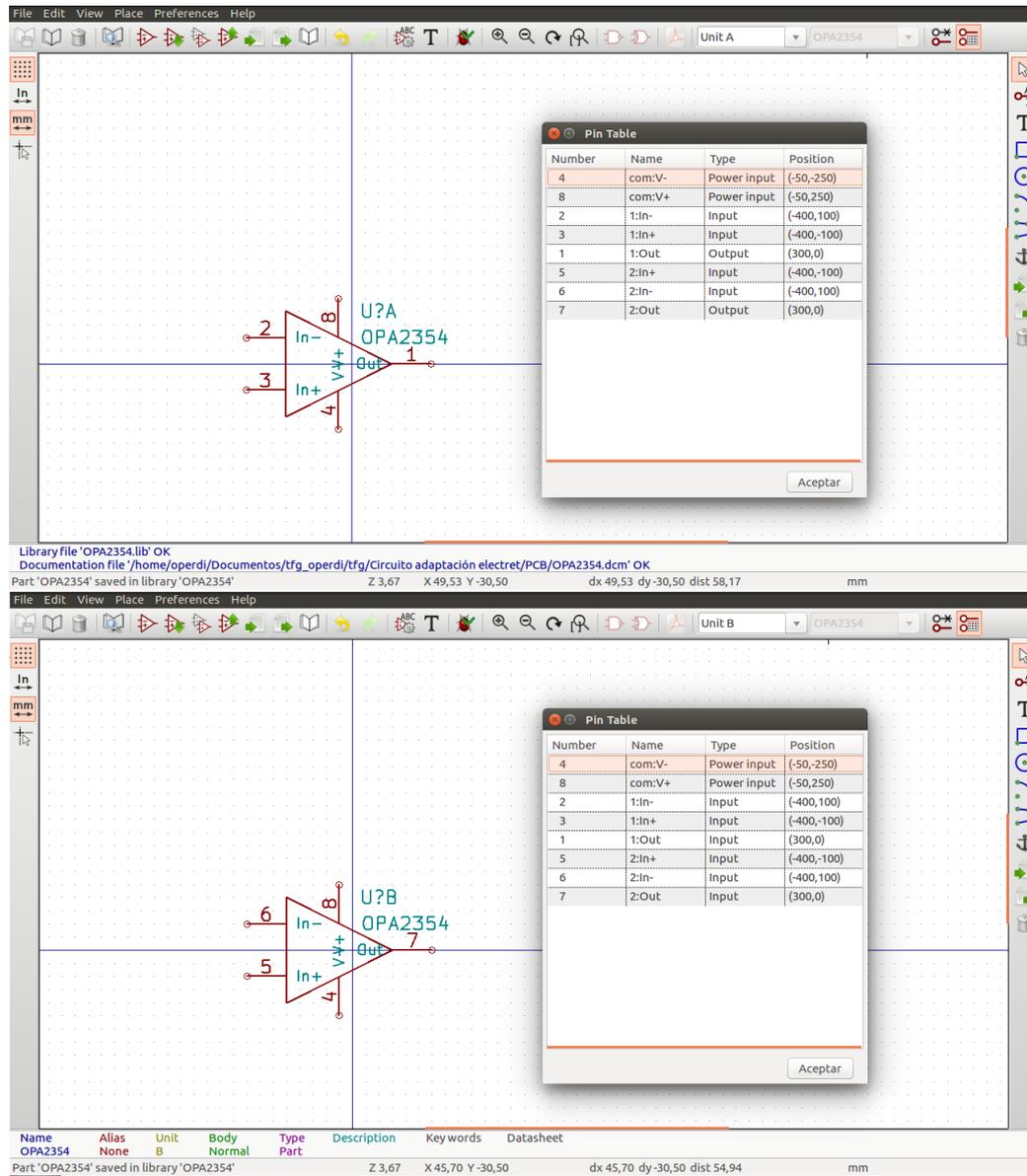


Figura 3.17: Integrados del OPA2354 y sus correspondientes pines.

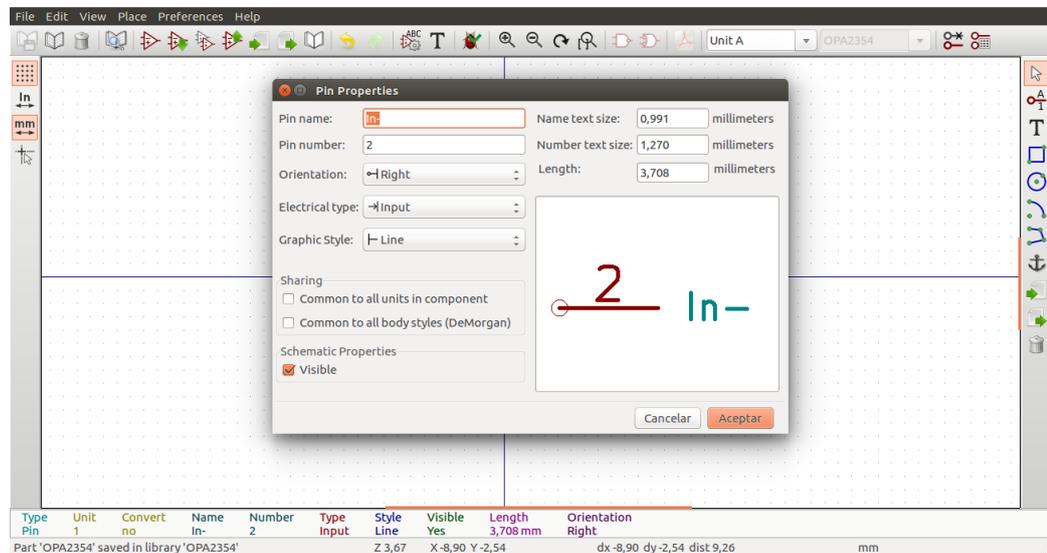


Figura 3.18: Edición de las características de un pin.

Teniendo en cuenta las especificaciones que se desean para la mecánica del diseño, se han medido diferentes elementos, como el conjunto de la Raspberry Pi y la tarjeta de audio, las placas de circuitos impresos y se han diseñado sus respectivas cajas, cuyos render se pueden observar en las figuras 3.19 y 3.20. El proceso de diseño en FreeCAD es muy variable, depende de la experiencia y preferencias del diseñador, en este caso se han utilizado el escenario *Part* que permite la creación de diferentes figuras y la realización de diferentes operaciones (unión, diferencia, intersección, corte, etc.) entre las mismas.

Teniendo estos elementos diseñados, se han incluido figuras para representar el resto de elementos, como tubos o elementos de soporte que se emplearán en la fabricación del diseño y se ha realizado un render del montaje completo del array, ver figura 3.21.

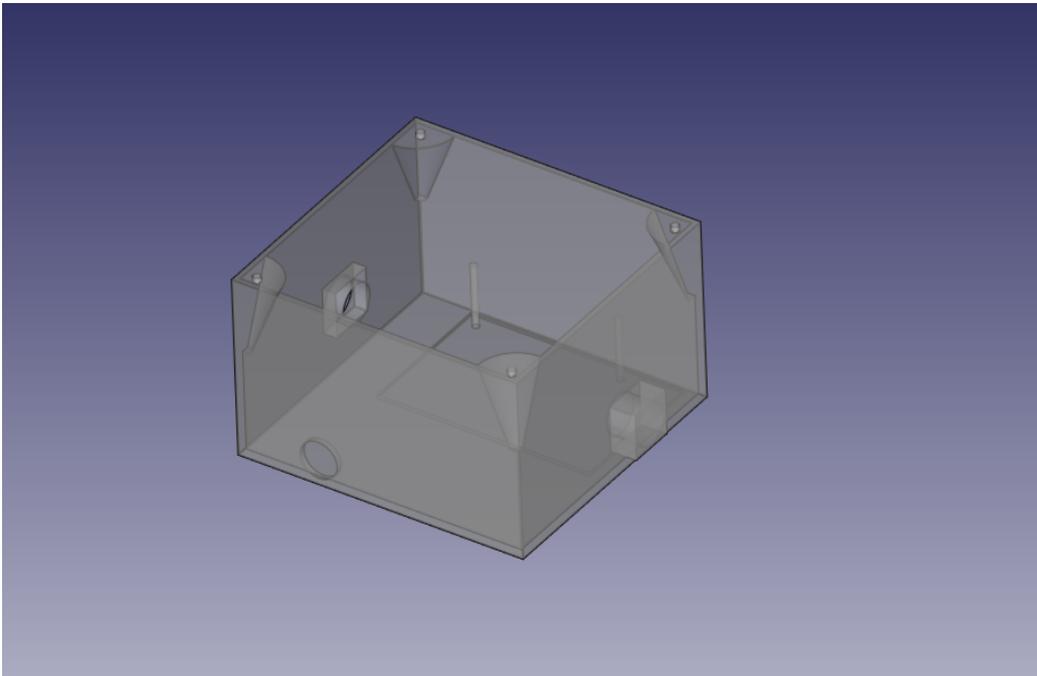


Figura 3.19: Render de la caja de la Raspberry Pi y la tarjeta de audio.

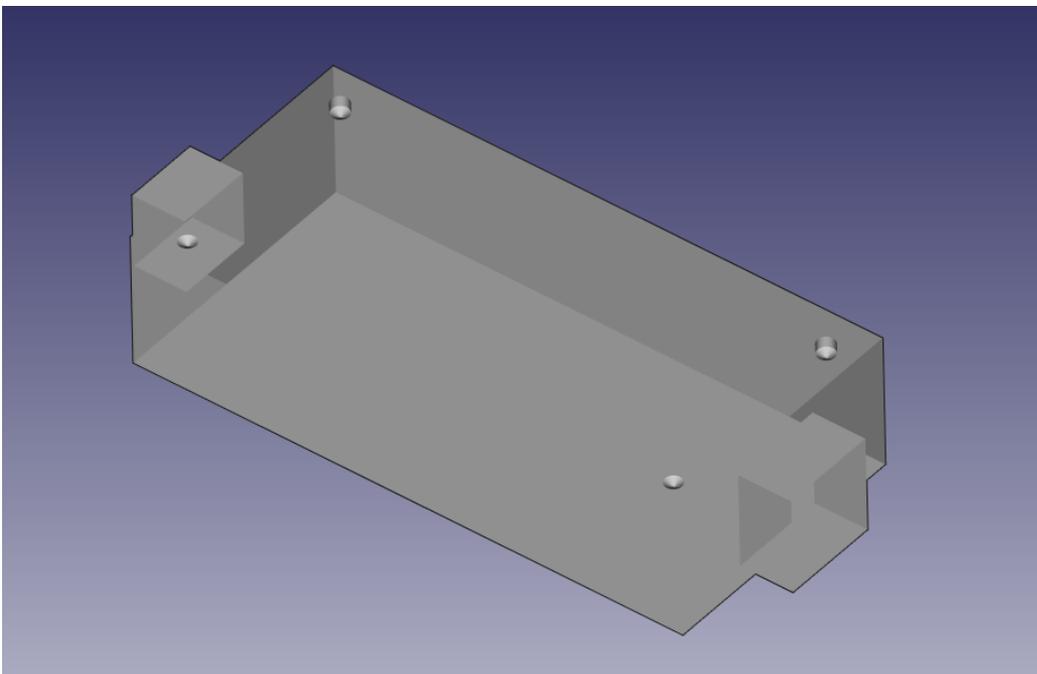


Figura 3.20: Render de la caja del amplificador.

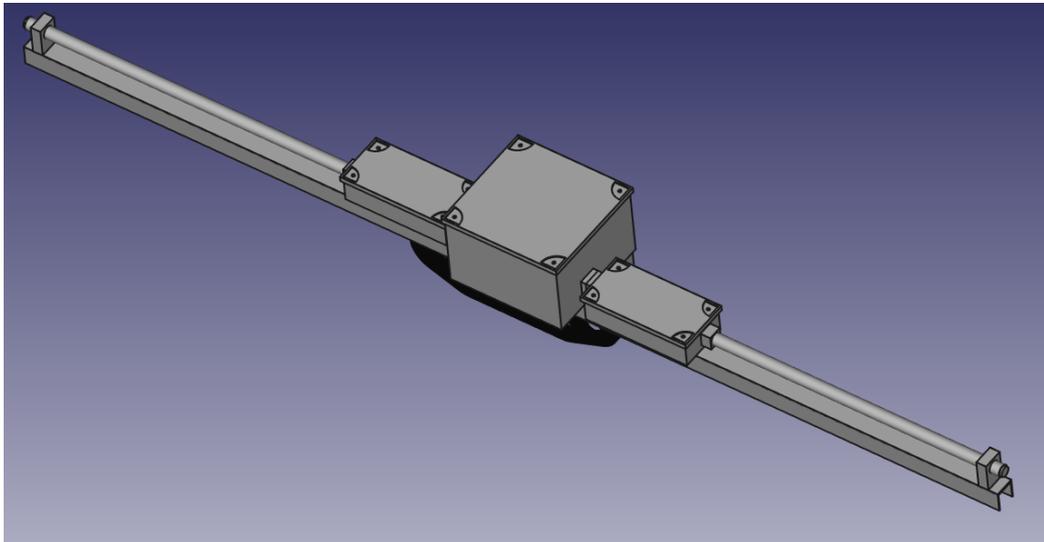


Figura 3.21: Render del diseño mecánico completo.

# Capítulo 4

## Fabricación

En este capítulo se exponen los procesos de fabricación empleados para obtener tanto la placa de circuitos impresos diseñada como los diferentes elementos que componen la parte mecánica.

Para ambos se han empleado los mecanismos de fabricación disponibles en el laboratorio, es decir, el atacado químico y horneado de la placa de circuitos impresos y la impresión 3D para los elementos mecánicos.

### 4.1. Placa de circuitos impresos

El primer paso para la fabricación de la placa es generar un archivo adecuado para poder generar el circuito de manera adecuada. En este caso se ha empleado un archivo SVG, que se ha impreso en papel satinado. Empleamos este papel pues la tinta no llega a penetrarlo, de modo que es posible colocar la impresión sobre la placa de cobre y plancharlo, quedando protegidas las áreas de cobre que queremos que tenga nuestra placa durante el atacado químico.

Tras imprimir el fotolito en el papel, se pega a la placa de cobre empleada y se plancha con la plancha de rodillos del laboratorio (figura 4.1). Para realizar el planchado correctamente se han de realizar múltiples pasadas, de modo que la tinta quede bien pegada al cobre. De lo contrario, al quitar el papel podrían quedar zonas desprotegidas en la placa en las que se quiera conservar el cobre. Para resolver este problema es posible rellenar las pistas o pads utilizando un rotulador y pintar sobre las zonas que queremos proteger. Para asegurar que todas las pistas han sido correctamente trazadas se pueden observar en la lupa (fig 4.2).

Una vez revisada la placa se procede al atacado químico de la misma, empleando para ello ácido clorhídrico. A través de este proceso las zonas de cobre que no están cubiertas por la tinta serán eliminadas. Una vez finalizado el atacado

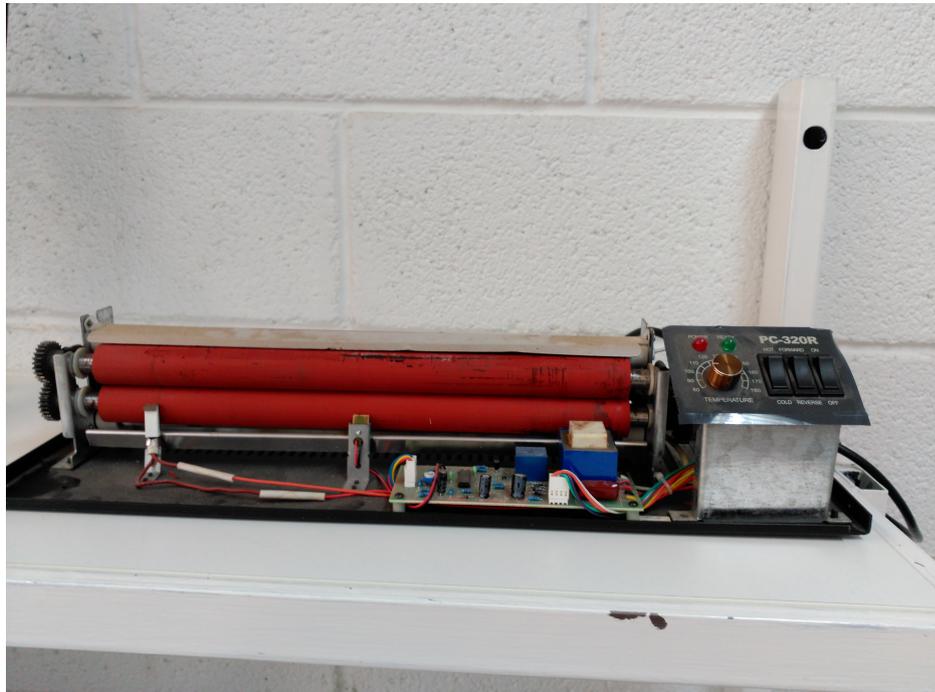


Figura 4.1: Plancha para las placas de circuitos impresos.

químico, se lava la placa para eliminar los restos de ácido y se elimina la tinta de la placa utilizando acetona para dejar el cobre al descubierto.

Una vez preparada la placa hay que colocar los componentes, para ello es posible soldarlos u hornear la placa. Se ha optado por el horneado de la placa y su posterior soldado en caso de que algunos componentes no queden debidamente colocados o conectados, eléctricamente, tras el primer proceso. Para soldar los componentes en el horneado se prepara una mezcla de estaño con plomo y flux que se coloca en los pads de la placa. Sobre la mezcla se colocan los componentes correspondientes y se procede al horneado.

Con el horno (figura 4.4) podemos elegir diferentes curvas de temperatura a las que someter la placa o imponer otra curva que deseemos. Realizamos el horneado con una curva que no tenga una temperatura demasiado elevada y que pueda quemar los componentes o la placa. Una vez finalizado el proceso se comprueban las soldaduras y las conexiones entre los componentes y los posibles cortocircuitos que se hayan podido producir. Si hay demasiados errores en la placa se añade flux a los pads y se hornea de nuevo la placa; en caso de haber pocos errores se desolda el componente y se suelda manualmente (figura 4.5), posteriormente se vuelven a comprobar las conexiones y se repite el proceso en caso de ser necesario.

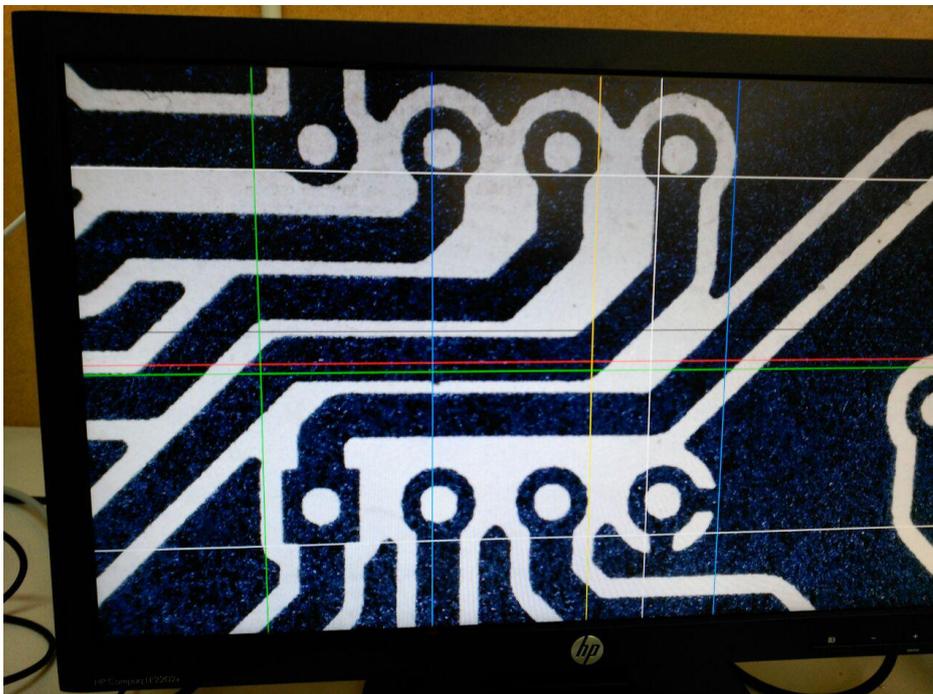


Figura 4.2: Revisión tras el planchado de una placa de circuitos impresos con la lupa.



Figura 4.3: Tanque para el atacado químico y lavado de la placa.



Figura 4.4: Horno para las placas de circuitos impresos.

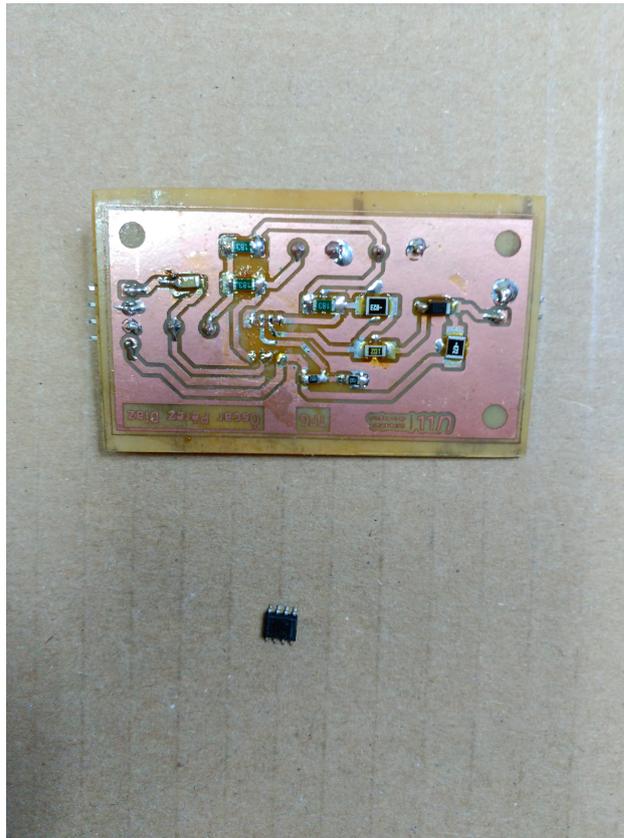


Figura 4.5: Operacional desoldado para su colocación.

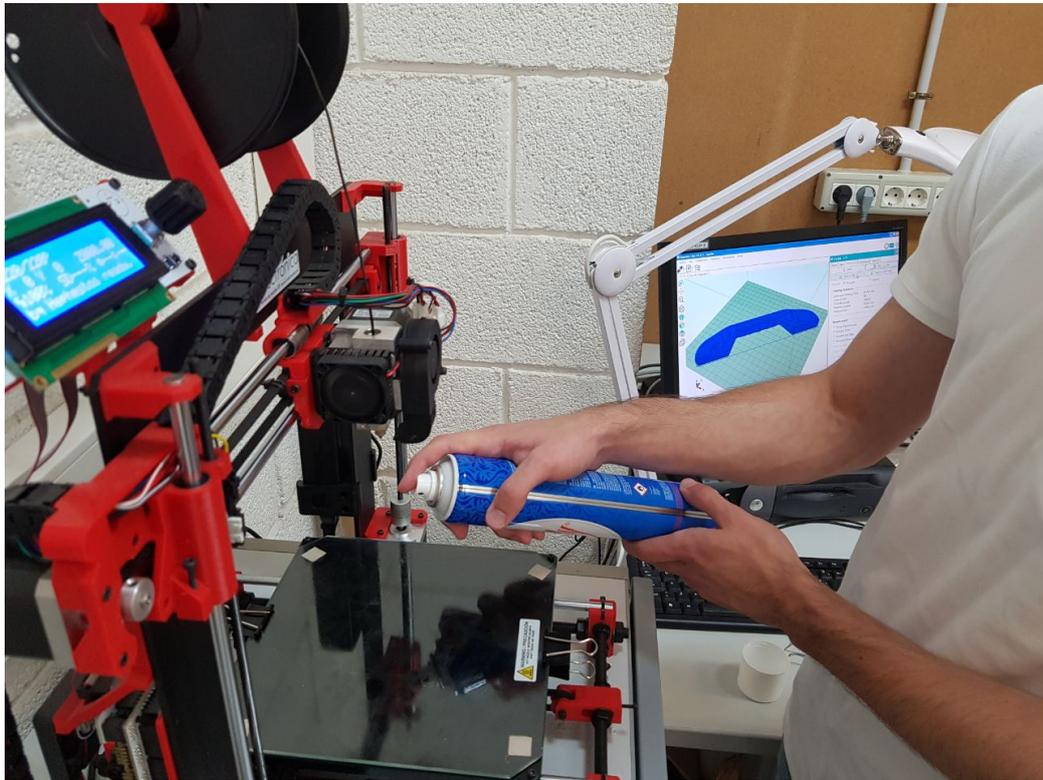


Figura 4.6: Preparado de la cama para la impresión de una pieza.

## 4.2. Elementos mecánicos

Los diferentes elementos mecánicos del array se han fabricado utilizando impresión 3D en poliácido láctico (PLA), un termoplástico procedente del almidón de maíz.

Tras generar los diferentes diseños se exportan en formato “.stl” para poder imprimirlos con la impresora 3D. Estos archivos se abren en el ordenador conectado a la impresora con el programa Repetier Host, donde podemos configurar los diferentes parámetros de la impresión como altura de la primera capa, altura de las capas, cantidad y tipo de relleno de la pieza, generación de material de soporte, velocidad de impresión o uso de *briming*, que es un borde exterior que se realiza para garantizar que el filamento está siendo expulsado cuando se imprime la pieza.

Una vez configurada la impresión se añade una capa de laca a la cama (figura 4.6), que favorece la sujeción de la pieza durante el proceso, y se revisa la impresión de las primeras capas para evitar que la pieza se despegue.

Finalmente, se retira la pieza de la cama y se eliminan el briming y otros elementos de sujeción de la pieza, obteniendo elementos fieles al diseño (ver figura

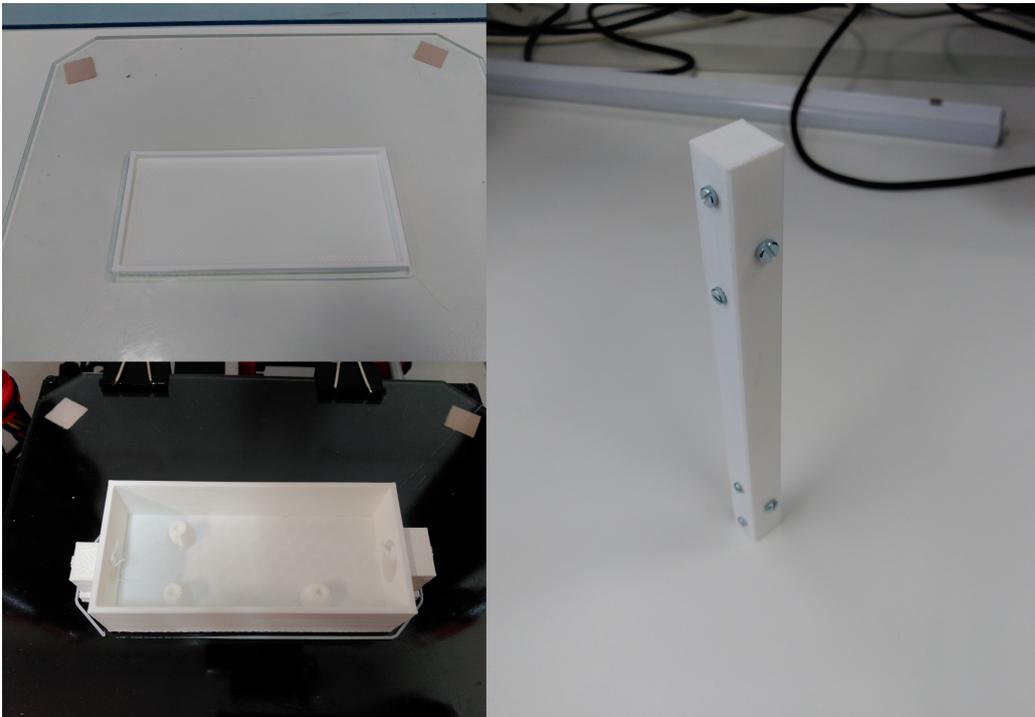


Figura 4.7: Algunas piezas impresas.

4.7). En el caso de algunas piezas, éstas se lijan para darles un mejor acabado o ajustar sus medidas, ya que la impresión tiene ciertas tolerancias.



# Capítulo 5

## Resultados y conclusiones

En resumen, los resultados obtenidos han sido los siguientes:

1. Se ha diseñado, tanto a nivel electrónico como mecánico, un prototipo de un array para la detección de fuentes sonoras. Empleando, además, *software* libre para dicho diseño.

2. Se ha diseñado un circuito electrónico para la adaptación de las señales. También se ha realizado el diseño y la fabricación de una placa de circuitos impresos, desarrollando además símbolos esquemáticos de componentes no existentes en la librería de símbolos.

3. Se han diseñado y fabricado los elementos necesarios para el montaje mecánico del array, permitiendo además su adaptación al medio acuático.

4. Se ha configurado correctamente la Raspberry Pi para adquirir la señal empleando la tarjeta de audio y permitir su uso a través de conexión WiFi.

5. Se han desarrollado diferentes programas para el análisis de la señal adquirida, obteniendo análisis de fourier, relación señal-ruido y estimación de la posición de la fuente sonora.

De modo que es posible concluir que los objetivos han sido alcanzados, generando un prototipo funcional del array.



# Capítulo 6

## *Results and conclusions*

*In summary, the results are the following:*

*1. It has been designed, both electronically and mechanically, a prototype of a microphone array for the detection of sound sources. Using free software for such design.*

*2. An electronic circuit has been designed for the adaptation of the signals. As well The design and manufacture of a printed circuit board, development in addition, schematic symbols of components that do not exist in the symbol library.*

*3. Different elements have been designed and manufactured for the mechanical assembly of the array, also allowing its adaptation to the aquatic environment.*

*4. The Raspberry Pi has been correctly configured to acquire the signal using the audio card and allow its use through WiFi.*

*5. Different programs have been developed for the analysis of the acquired signal, obtaining Fourier analysis, signal-to-noise ratio and position estimation of the sound source.*

*So it is possible to conclude that the objectives have been achieved, generating a functional prototype of the array.*



# Bibliografía

- [1] *About FreeCAD - FreeCAD Documentation*. URL: [https://www.freecadweb.org/wiki/About\\_FreeCAD](https://www.freecadweb.org/wiki/About_FreeCAD) (visitado 27-06-2018).
- [2] Daniel T. Blumstein<sup>1</sup>, Daniel J. Mennill, Patrick Clemins, Lewis Girod , Kung Yao, Gail Patricelli, Jill L. Deppe, Alan H. Krakauer, Christopher Clark, Kathryn A. Cortopassi, Sean F. Hanser, Brenda McCowan, Andreas M. Ali and Alexander N. G. Kirschel. «Acoustic monitoring in terrestrial environments using microphone arrays: applications, technological considerations and prospectus». En: *Journal of Applied Ecology* (2011).
- [3] *Getting Started in KiCad*. URL: [http://docs.kicad-pcb.org/4.0.5/en/getting\\_started\\_in\\_kicad.html#make-schematic-components-in-kicad](http://docs.kicad-pcb.org/4.0.5/en/getting_started_in_kicad.html#make-schematic-components-in-kicad) (visitado 07-08-2018).
- [4] Allan Hambley. *ELECTRÓNICA 2ED*. 2.<sup>a</sup> ed. Madrid, España: Pearson, 2001. ISBN: 9788420529998.
- [5] *KiCad EDA*. en-us. URL: <http://kicad-pcb.org/> (visitado 26-06-2018).
- [6] *LTspice | Design Center | Analog Devices*. URL: <http://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html> (visitado 16-08-2018).
- [7] *Secure Shell*. es. Page Version ID: 105565950. Feb. de 2018. URL: [https://es.wikipedia.org/w/index.php?title=Secure\\_Shell&oldid=105565950](https://es.wikipedia.org/w/index.php?title=Secure_Shell&oldid=105565950) (visitado 01-07-2018).
- [8] *SSH (Secure Shell) - Raspberry Pi Documentation*. URL: <https://www.raspberrypi.org/documentation/remote-access/ssh/> (visitado 01-07-2018).
- [9] *Welcome to Python.org*. en. URL: <https://www.python.org/about/> (visitado 27-06-2018).
- [10] Robin C. Whytock y James Christie. «Solo: an open source, customizable and inexpensive audio recorder for bioacoustic research». En: *Methods in Ecology and Evolution* 8 (oct. de 2016), págs. 308-312.





**Universidad  
de La Laguna**

**Escuela Superior  
de Ingeniería y Tecnología**

**Departamento de Ingeniería Industrial**

**ESCUELA SUPERIOR DE INGENIERÍA  
Y TECNOLOGÍA**

## **Trabajo de Fin de Grado**

**CETARRAY: Un array de hidrófonos inteligentes  
basado en una red de Raspberry Pi para la detección  
pasiva de cetáceos.**

**TOMO II**

## **Pliego de condiciones y presupuesto**

**Titulación:** Grado en Ingeniería Electrónica Industrial y  
Automática

**Estudiante:** Óscar Pérez Díaz

**Tutor:** Fernando Luis Rosa González  
1 de septiembre de 2018



# Capítulo 7

## Pliego de condiciones

La condición principal para la realización del proyecto ha sido el uso de la Raspberry Pi como controlador y el uso de programas de *software* libre para el diseño de las diferentes partes del proyecto:

- Hacer uso de Git como herramienta para el control de versiones y la compartición de archivos.
- Utilizar kiCad para el diseño de la placa de circuitos impresos.
- Emplear FreeCad para el diseño de la mecánica del array.
- Redactar y generar la documentación del proyecto utilizando *Kile*.



# Capítulo 8

## Presupuesto

El presupuesto del proyecto alcanza los 5981.57 €. Es posible ver la descomposición de dicha cantidad en la tabla 8.1 donde se especifica cada elemento que compone el presupuesto, su distribuidor y su cantidad, precio unitario y precio total. Para las horas de diseño se ha establecido un precio de 20 €/hora y 18 €/hora para las horas de fabricación, ambos precios incluyen el coste del trabajo y costes de Seguridad Social e impuestos.

El coste de los diferentes componentes empleados en el diseño del circuito se desarrolla en la tabla 8.2, que incluye la referencia del fabricante de cada componente, su valor correspondiente, la cantidad y el precio por unidad, así como el precio total. De igual manera, en la tabla 8.3 se exponen los costes de los diferentes elementos utilizados en la fabricación mecánica del array.

Elemento	Distribuidor	Uni.	€/ud	€
Raspberry Pi 3B	RS Components	1	35,66	35,66
Cirrus Logic Audio Card	Farnell element14	1	34,18	34,18
Componentes electrónicos	DigiKey	28	-	9,96
Placa de circuitos impresos	RS Components	1	4,16	4,16
Elementos mecánicos	Leroy Merlin	6	-	42,66
Batería	Leroy Merlin	1	14,95	14,95
Precio por hora de diseño	-	220	20	4400
Precio por hora de fabricación	-	80	18	1440

Tabla 8.1: Presupuesto. **Uni.:** Unidad, **€/ud:** Precio por unidad, **€:** Total en euros.

Grupo	Elemento	Valor	Uni.	€/ud	€
Resistencias	ERA-6AEB472V	4,7 kOhm	2	0,31	0,62
	RR1220P-103-D	10 kOhm	4	0,09	0,36
	RR1220P-102-D	1 kOhm	2	0,09	0,18
	CRGCQ1210F82K	82 kOhm	2	0,19	0,38
	ERJ-14YJ183U	18 kOhm	6	0,19	1,14
Condensadores	K101J15C0GF5TL2	100 pF	6	0,22	1,32
	C0805C224K3RACTU	0,22 uF	2	0,15	0,30
	F921C105MPA	1 uF uF	2	0,33	0,66
Operacionales	OPA2354AIDDAR	-	2	2,5	5

Tabla 8.2: Precio detallado de los componentes electrónicos. **Uni.:** Unidad, **€/ud:** Precio por unidad, **€:** Total en euros.

Elemento	Uni.	€/ud	€
Filamento PLA	1	19,95	19,95
Tubos y barra	3	2,20	6,60
Tornillos	2	3,58	7,16
Pegamento	1	8,95	8,95

Tabla 8.3: Precio detallado de los elementos mecánicos. **Uni.:** Unidad, **€/ud:** Precio por unidad, **€:** Total en euros.



**Universidad  
de La Laguna**

**Escuela Superior  
de Ingeniería y Tecnología**

**Departamento de Ingeniería Industrial**

**ESCUELA SUPERIOR DE INGENIERÍA  
Y TECNOLOGÍA**

## **Trabajo de Fin de Grado**

**CETARRAY: Un array de hidrófonos inteligentes  
basado en una red de Raspberry Pi para la detección  
pasiva de cetáceos.**

**TOMO III**

**Anexos**

**Titulación:** Grado en Ingeniería Electrónica Industrial y  
Automática

**Estudiante:** Óscar Pérez Díaz

**Tutor:** Fernando Luis Rosa González

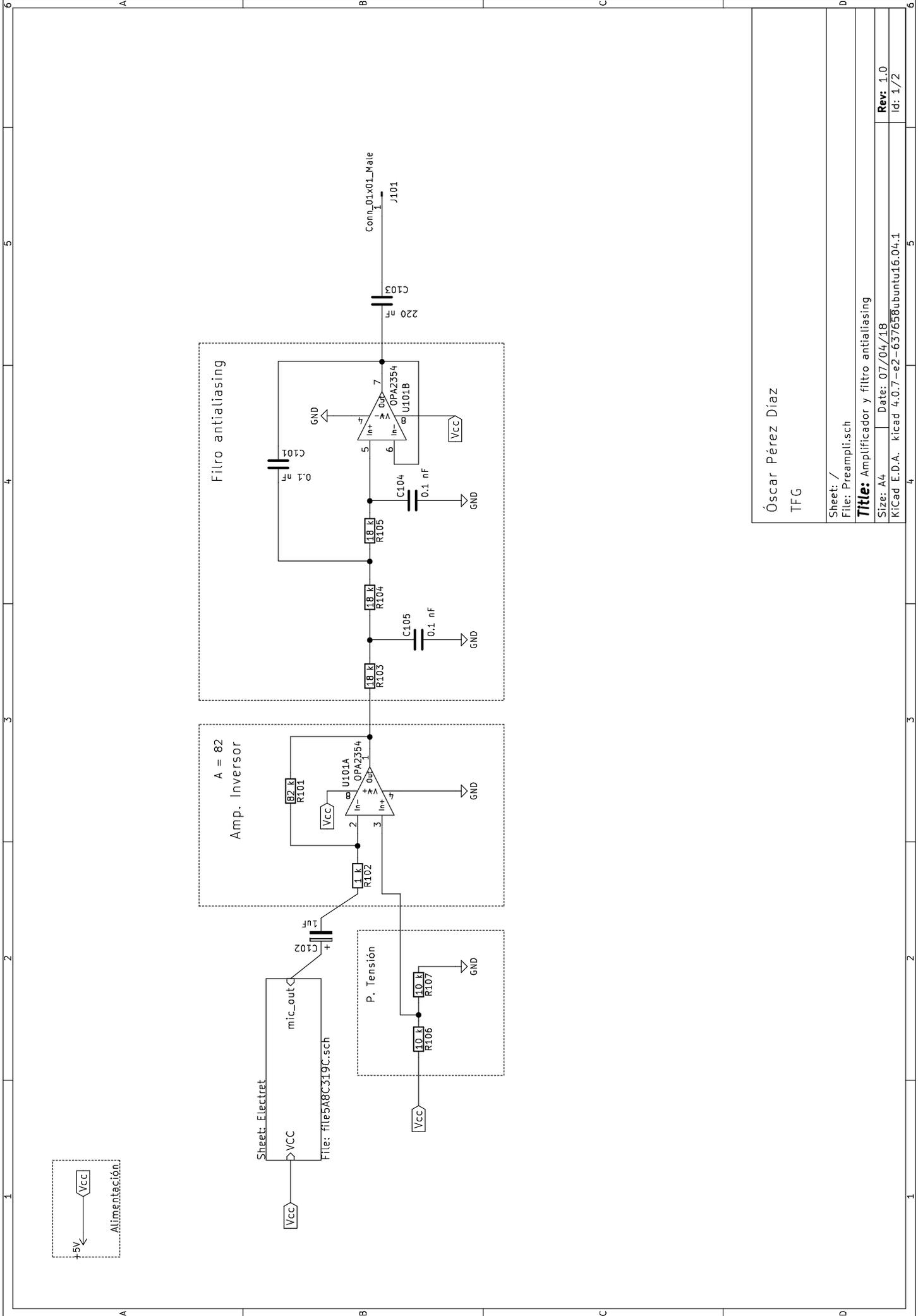
1 de septiembre de 2018



# **Capítulo 9**

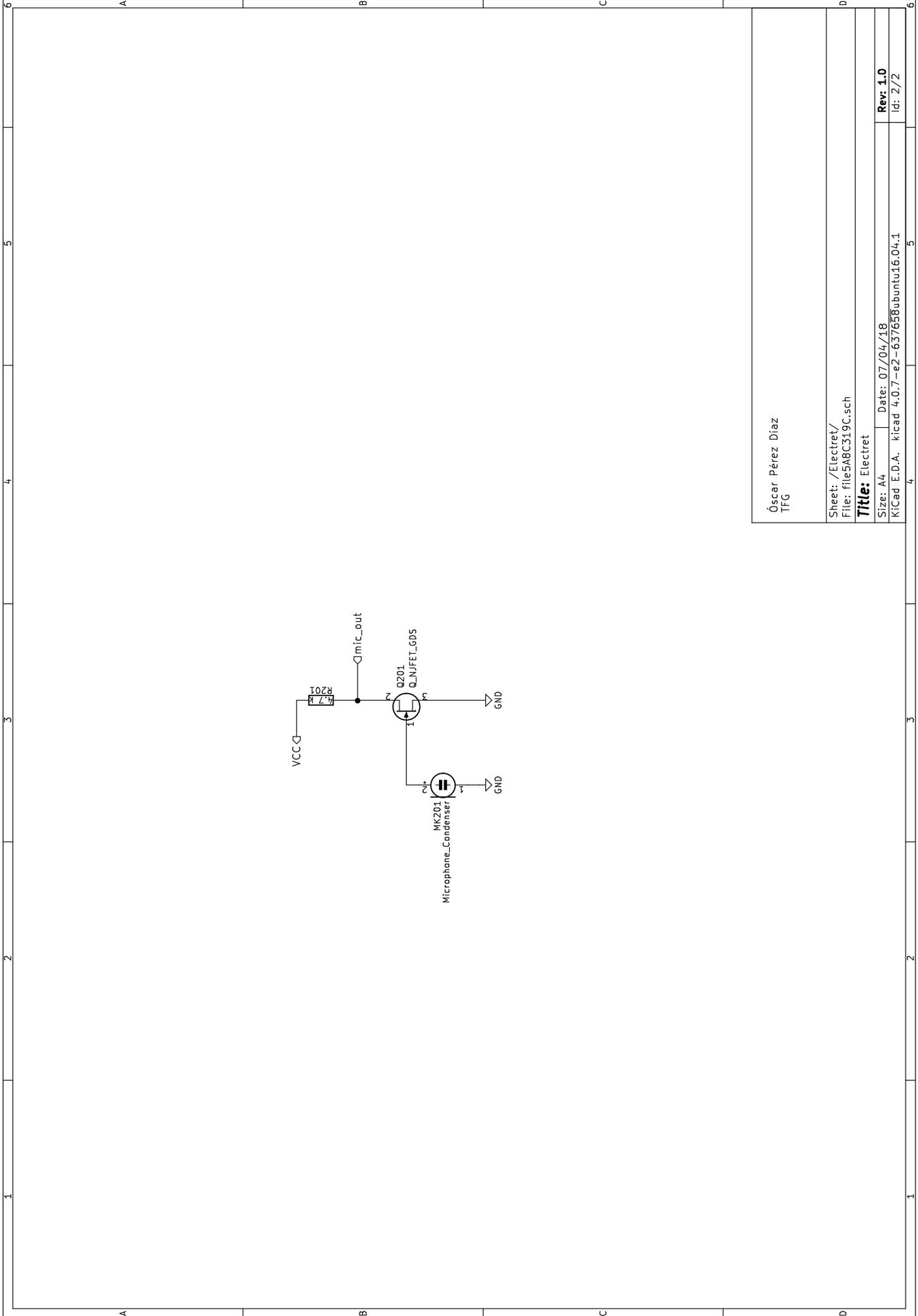
## **Planos**

### **9.1. Etapa de adaptación**



Óscar Pérez Díaz  
TFG

Sheet: /  
File: Preampli.sch  
**Title:** Amplificador y filtro anti-aliasing  
Size: A4 Date: 07/04/18  
KICad E.D.A. kicad 4.0.7-e2-637658ubuntu16.04.1  
Rev: 1.0  
Id: 1/2



Óscar Pérez Díaz  
TFG

Sheet: /Electret/  
File: file5A8C319C.sch

**Title:** Electret

Size: A4 | Date: 07/04/18

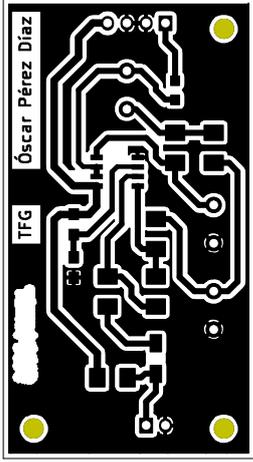
KiCad E.D.A. kicad 4.0.7-e2-637658ubuntu16.04.1

**Rev. 1.0**

Id: 2/2

## 9.2. Placa de circuitos impresos

1	A	B	C	D
2	A	B	C	D
3	A	B	C	D
4	A	B	C	D
5	A	B	C	D
6	A	B	C	D



Óscar Pérez Díaz  
TFG

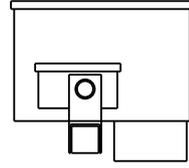
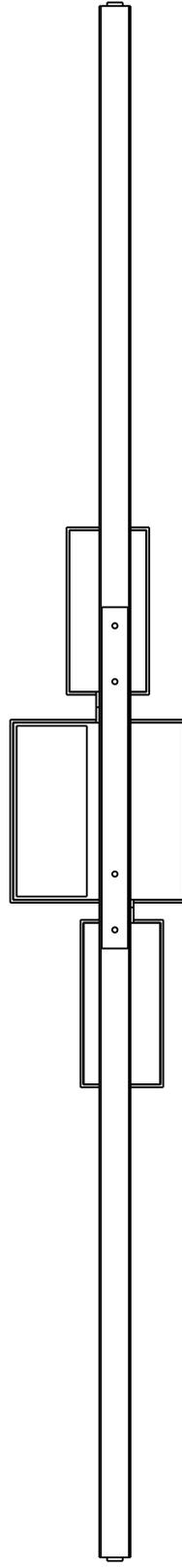
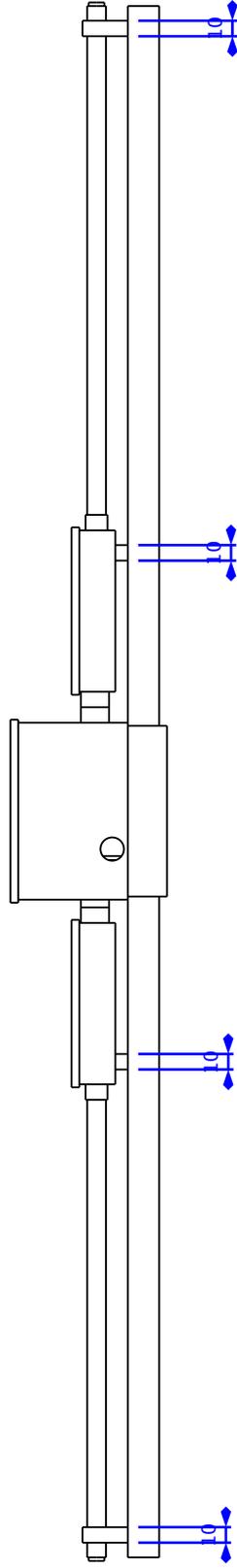
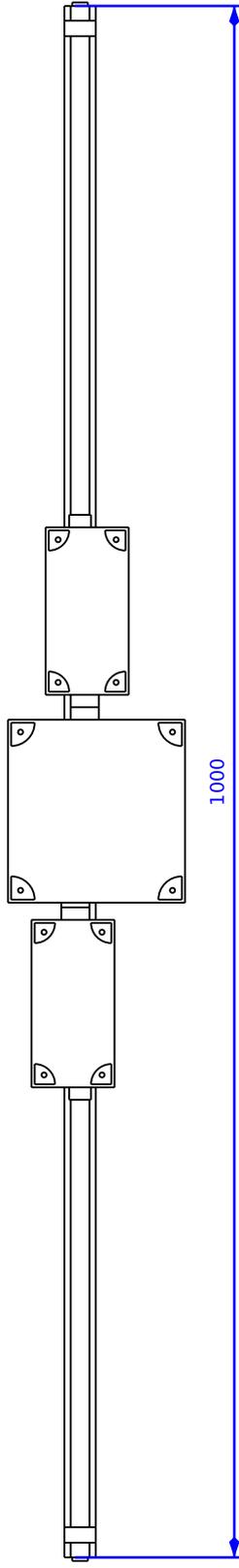
Sheet:  
File: Preimpli.kicad\_pcb

**Title: Diseño placa de circuitos impresos**

Size: A4	Date:
KiCad E.D.A. kicad 4.0.7-e2-637658ubuntu16.04.1	Id: 1/1

Rev. 3.0  
Id: 1/1

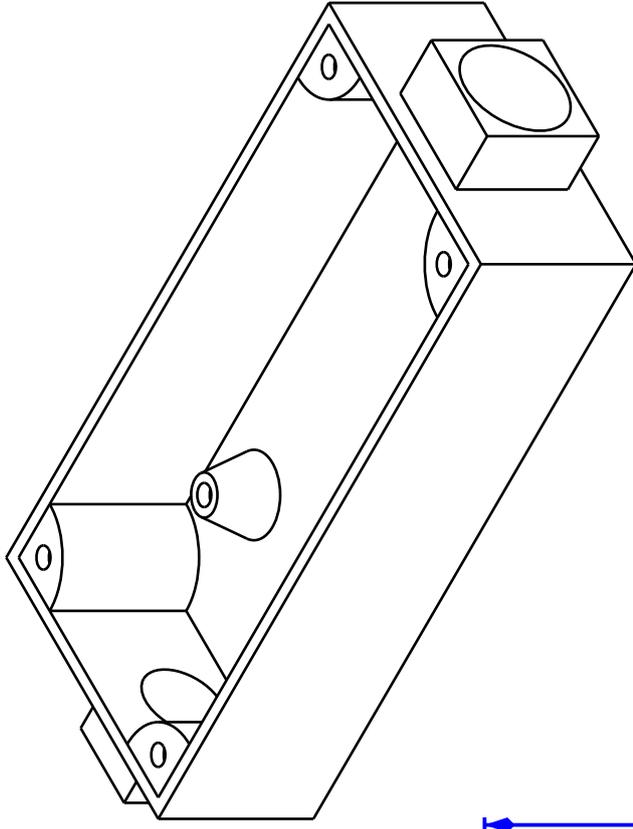
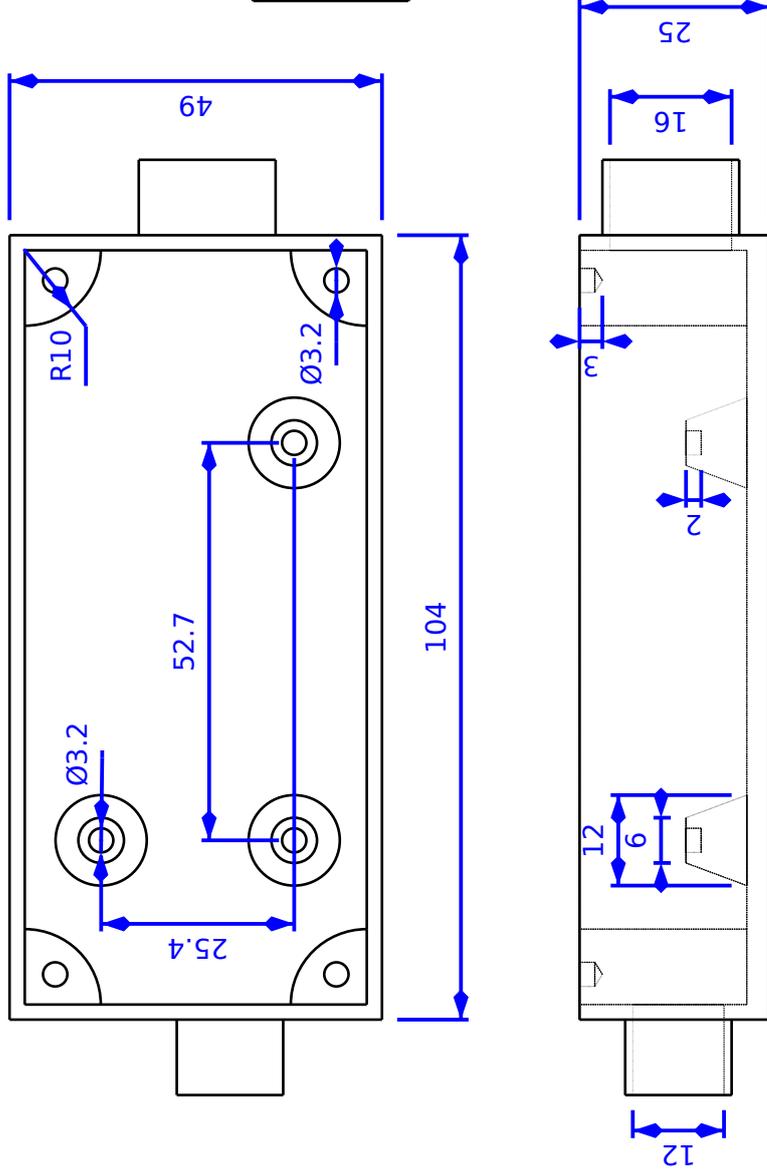
### **9.3. Elementos mecánicos**



Created by: <b>Oscar Pérez Díaz</b>	Title: <b>Montaje mecánico general</b>	
Supplementary information:		
Size: <b>A3</b>	Sheet: <b>1 / 7</b>	Scale: <b>0.3</b>
Part number: <b>PN</b>		Drawing number: <b>DN</b>
Date: <b>18/06/2018</b>		Revision: <b>REV 3</b>

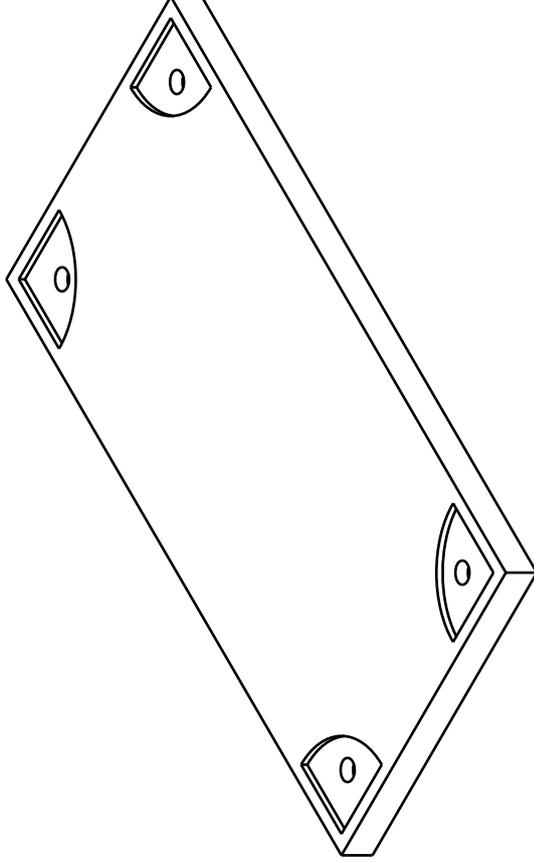
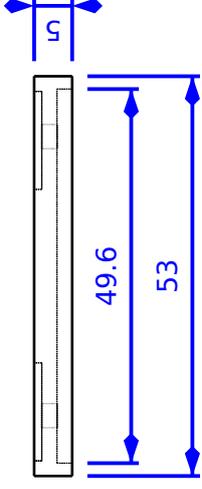
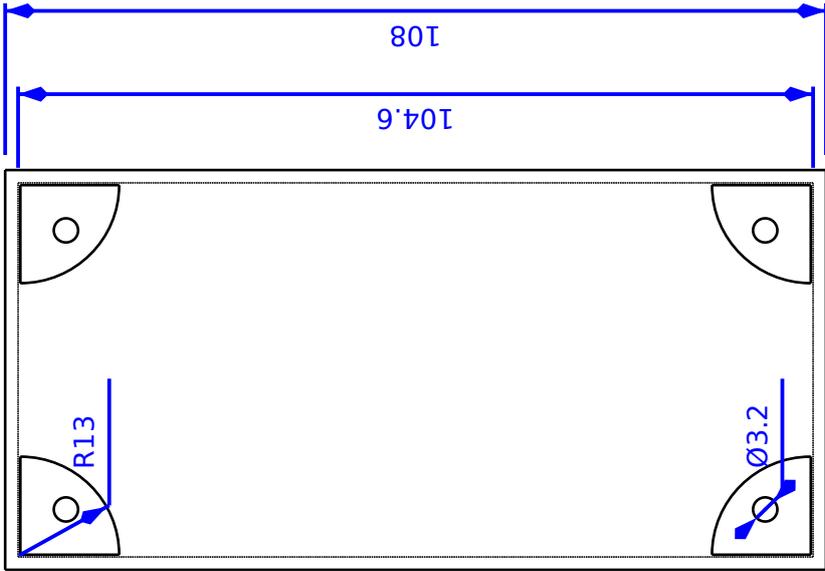
**FreeCAD DRAWING**  
**Medidas en mm**  
**TFG**



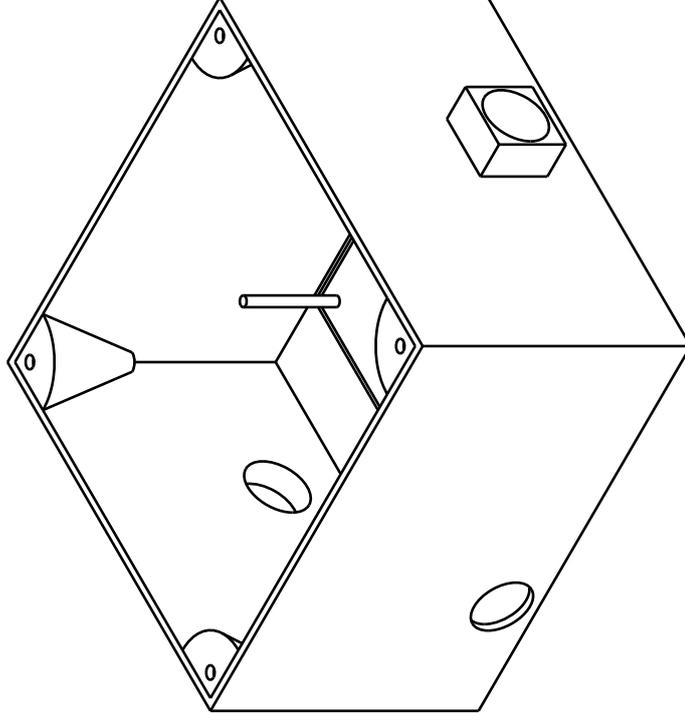
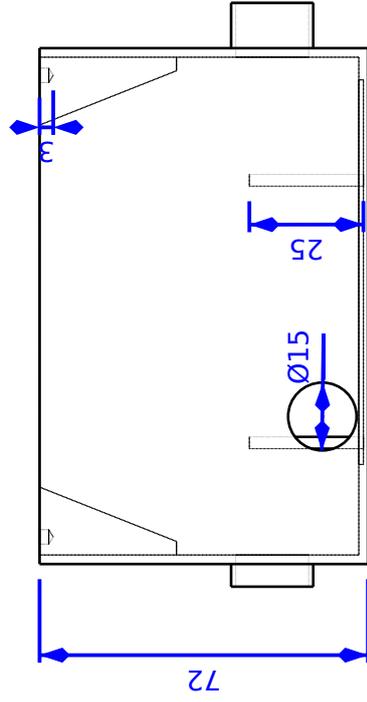
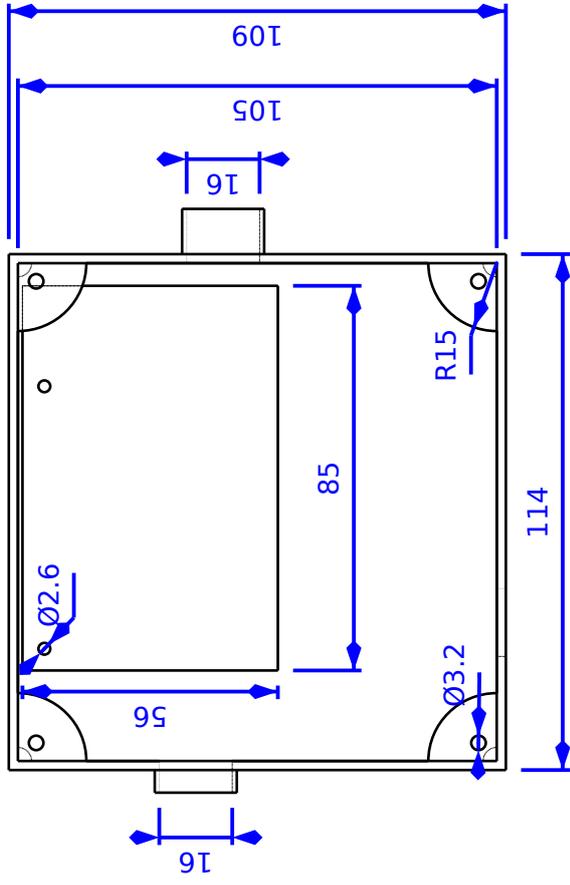


Created by: <b>Óscar Pérez Díaz</b>	Title: <b>Caja amplificador</b>	Size: <b>A4</b>	Sheet: <b>2 / 7</b>	Scale: <b>1</b>
Supplementary information: <b>FreeCAD DRAWING</b> <b>Unidades en mm</b> <b>TFG</b>		Part number: <b>PN</b>	Drawing no.: <b>DN</b>	Revision: <b>REV 2</b>
		Date: <b>18/06/2018</b>		Revision: <b>REV 2</b>





Created by: <b>Óscar Pérez Díaz</b>	Title: <b>Tapa caja amplificador</b>	Size: <b>A4</b>	Sheet: <b>3 / 7</b>	Scale: <b>1</b>
Supplementary information: <b>FreeCAD DRAWING</b> Unidades en mm <b>TFG</b>		Part number: <b>PN</b>	Drawing no.: <b>DN</b>	Revision: <b>REV 2</b>
			Date: <b>18/06/2018</b>	Revision: <b>REV 2</b>



Created by:

Óscar Pérez Díaz

Supplementary information:

FreeCAD DRAWING

Unidades en mm

TFG

Title:

Caja Raspberry Pi

Size:

A4

Sheet:

4 / 7

Scale:

0.6

Part number:

PN

Drawing no.:

DN

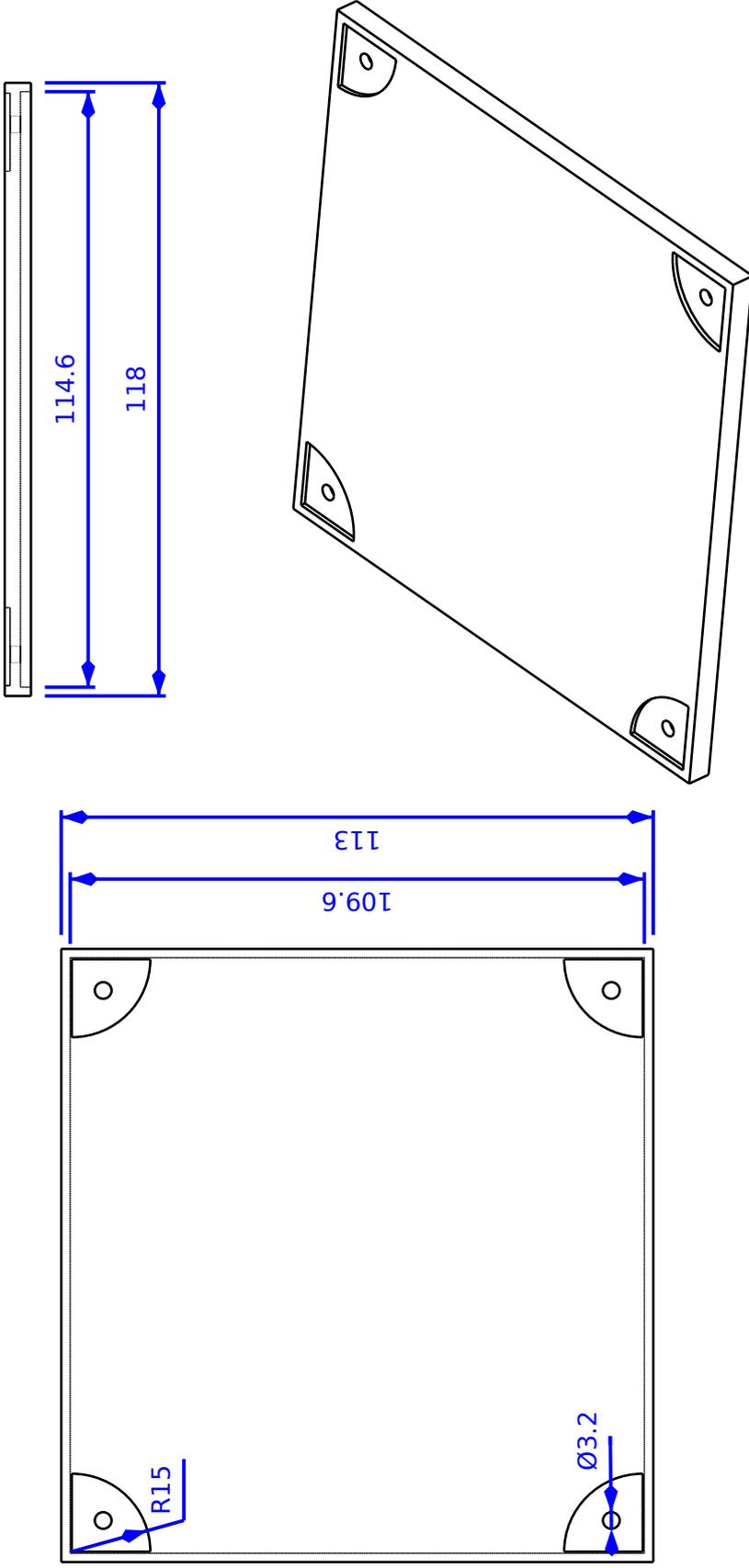
Date:

18/06/2018

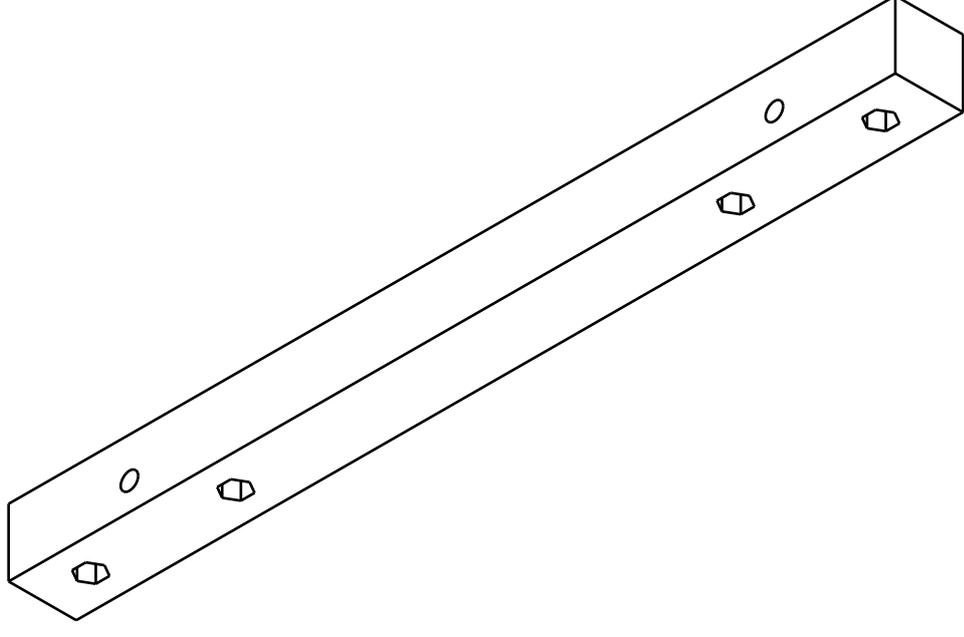
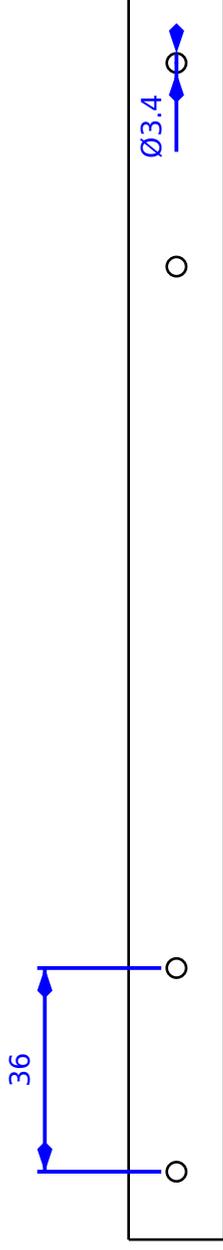
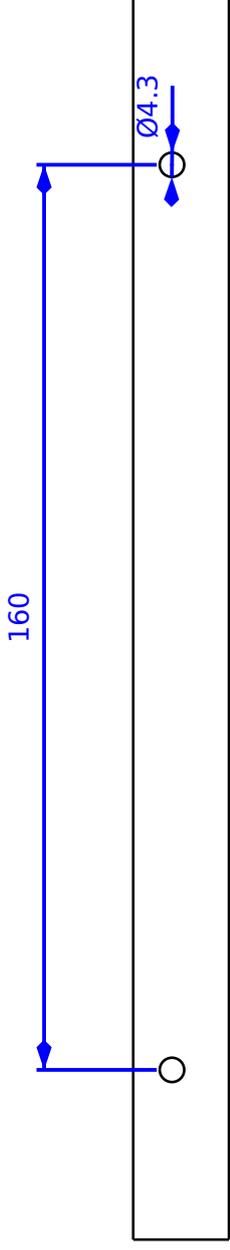
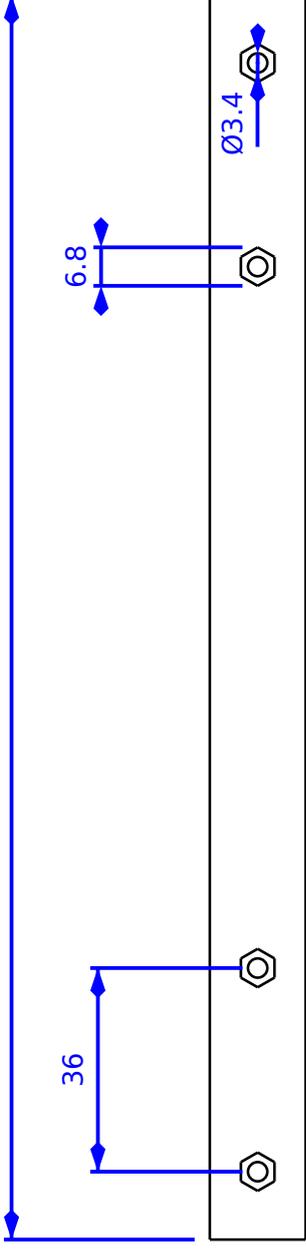
Revision:

REV 1



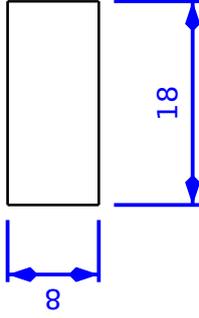
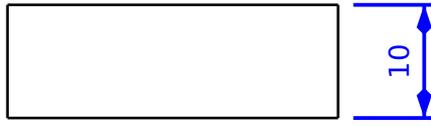
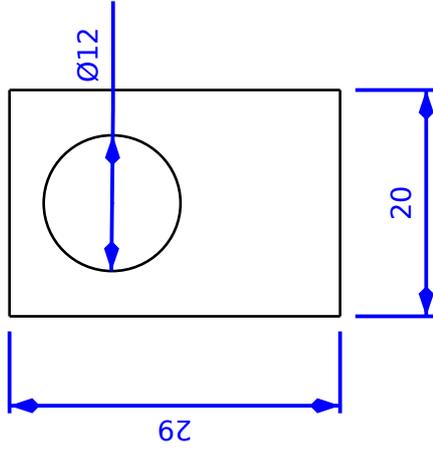
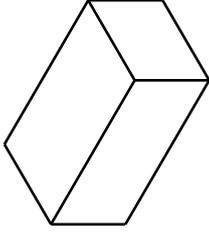
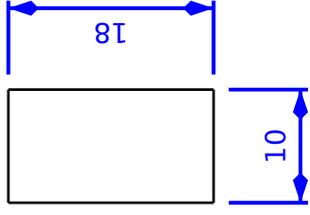
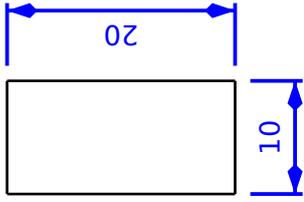
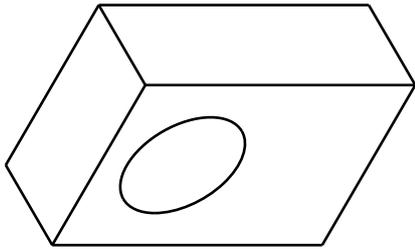


Created by: <b>Óscar Pérez Díaz</b>	Title: <b>Tapa Raspberry</b>	Size: <b>A4</b>	Sheet: <b>5 / 7</b>	Scale: <b>0.75</b>
Supplementary information: <b>FreeCAD DRAWING</b> Unidades en mm		Part number: <b>PN</b>	Drawing no.: <b>DN</b>	Revision: <b>REV 2</b>
<b>TFG</b>		Date: <b>18/06/2018</b>		Revision: <b>REV 2</b>



Created by: <b>Óscar Pérez Díaz</b>	Title: <b>Soporte asa</b>	Size: <b>A4</b>	Sheet: <b>6 / 7</b>	Scale: <b>0.75</b>
Supplementary information: <b>FreeCAD DRAWING</b> Unidades en mm <b>TFG</b>		Part number: <b>PN</b>		Drawing no.: <b>DN</b>
		Date: <b>18/06/2018</b>	Revision: <b>REV 1</b>	





Created by:

Óscar Pérez Díaz

Supplementary information:

FreeCAD DRAWING

Undades en mm

TFG

Title: Elementos de soporte

Size: A4

Sheet: 7 / 7

Scale: 1.5

Part number: PN

Drawing no.: DN

Date:

18/06/2018

Revision:

REV 1





# Capítulo 10

## *Datasheets*

### 10.1. *Cirrus Logic Audio Card*

# Cirrus Logic Audio Card User Documentation

## Contents

1. An introduction to the Cirrus Logic Audio Card for Raspberry Pi
2. Features
3. How to install the card to your to Raspberry Pi
4. Installing software to run on Raspberry Pi
5. Getting started with audio from your Cirrus Logic Audio Card
6. Soundcard usage scenarios
7. Annexe
  - a. Datasheet of WM5102 audio CODEC
  - b. Datasheet of WM8804 SPDIF receiver/transmitter
  - c. Datasheet of WM7220 Digital Microphone Module
  - d. Schematic diagram of soundcard
  - e. Feature header signal assignment
  - f. Performance
  - g. Electrical architecture of the soundcard

## 1. Introduction to Cirrus Logic Audio Card for Raspberry Pi

Raspberry Pi, whilst being equipped with audio capability, remains limited in a number of ways. The limitations are intended to maintain Raspberry Pi's low price point, but still represent a limitation to users interested in exploring the audio capability of Raspberry Pi.

In terms of audio, there are no ways to capture audio using Raspberry Pi alone, and audio output is limited to two paths; analogue, via its onboard 3.5 mm stereo output jack, and digital, via its onboard HDMI output. Whilst the HDMI output provides the potential for high quality rendering of audio (depending on what HDMI devices are used to finally convert audio from its digital format to an analogue signal), the audio quality from the analogue 3.5 mm stereo output jack is universally recognised as being of an 'acceptable' quality level – no more.

The most important limitation is Raspberry Pi's lack flexibility in terms of multiple types of audio input sources, and outputs.

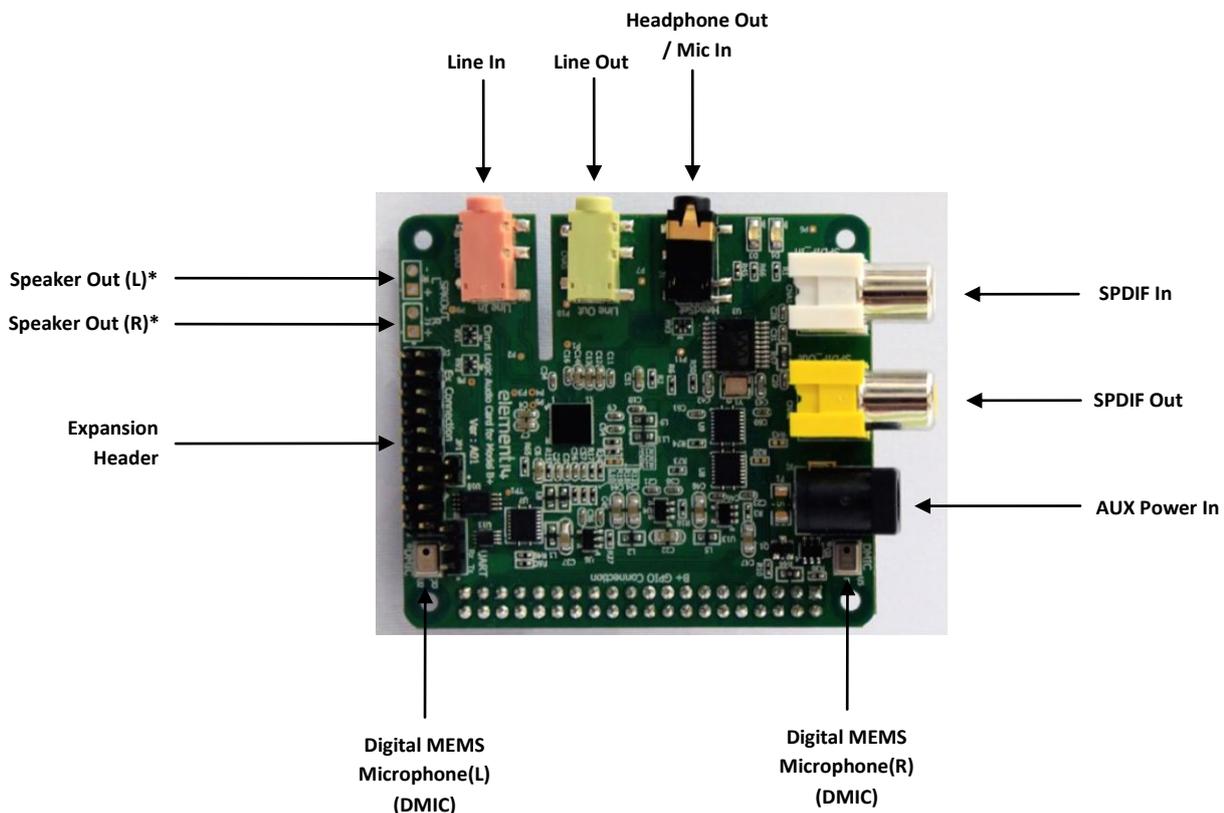
Early in 2014, Cirrus Logic acquired Wolfson Microelectronics, and Raspberry Pi released the new 40-pin GPIO featured, for example featured on models B+ and A+. In addition to the new GPIO configuration, the 8-pin P5 connector was also dropped, allowing for an altogether easier interconnect experience for accessories. The Cirrus Logic Audio card is therefore a replacement for the Wolfson Audio Card, and addresses the need for a high quality audio solution for 40-pin versions of Raspberry Pi. It contains the same core silicon and therefore has the same high performance as its predecessor, but has been improved in a number of ways.

## 2. Features

This product, designed by element14 and Cirrus Logic in partnership, addresses the above by providing a rich set of high quality audio features, including the following:

### Features

- Compatible with Raspberry Pi B+ and A+ onwards (with 40-pin extended GPIO, and no P5 connector)
- Analogue line-level output and input
- Digital stereo audio input and output (SPDIF)
- High quality headphone output, with microphone facility (for headphones with boom microphone)
- Onboard stereo digital microphones based on MEMS technology
- Ability to render High Definition (HD) Audio
- Arrives bundled with five High Definition (HD) audio files to demonstrate the systems capability.
- Onboard power amplifier for directly driving loudspeakers. (Requires headers to be fitted.<sup>1</sup>)
- Expansion header to allow connection to host boards other than Raspberry Pi.
- Back power protection, allows Raspberry Pi to be powered from the Cirrus Logic Audio card, allowing for convenient set up.
- New universal software support for both this card, and the former Wolfson Audio Card.



**Diagram 1: Connections to and from the Cirrus Logic Audio Card**

#	Audio Function	Script to run
1.	Play an audio file to your headphones	Playback_to_Headset.sh
2.	Play an audio file to the line out jack	Playback_to_Lineout.sh
3.	Play an audio file to loudspeakers	Playback_to_Speakers.sh
4.	Play an audio file to the SPDIF output of your card	SPDIF_playback.sh
5.	Record from onboard digital microphones	Record_from_DMIC.sh
6.	Record from the boom microphone of your gamer/VoIP headphones	Record_from_Headset.sh
7.	Record from the analogue Line In socket	Record_from_lineIn.sh
9.	Record from the SPDIF input of your card	SPDIF_record.sh
10.	Reset all audio paths	Reset_paths.sh
11.	Record from line in socket at microphone sensitivity level, and with mic bias voltage switched ON	Record_from_lineIn_Micbias.sh

**Table 1: A list of the Use Case Scripts for setting up the audio card.**

As an example, let us say that you want to play one of the bundled High Definition audio tracks that are included in the software package, and you want to play it to your headset. In this case, and at the command line in **LXTerminal**, type the following command:

```
./Playback_to_Headset.sh <Enter>
```

<b>Line Output</b>			
<b>Parameter</b>	<b>Description/Conditions</b>	<b>Typical Value</b>	<b>Units</b>
Connector	Electrical output via 3 pole 3.5mm Socket		
Rout	Output impedance	16	Ohms
Cload	Max capacitive load on output	2	nF
Rload		10	kOhms
Vout	Full scale output signal level	1	Vrms
Vnoise	Noise Floor, no signal applied, A-weighting filter	4.5	uVrms
THD	Total Harmonic Distortion	0.005	%
Notes	Test conditions as follows unless otherwise noted: 48kHz sample rate, test signal 1kHz sine wave, bandwidth measured 20Hz to 20kHz, Rload = 10kOhms		

<b>Line Input</b>			
<b>Parameter</b>	<b>Description</b>	<b>Typical Value</b>	<b>Units</b>
Connector	Electrical input via 3 pole 3.5mm Socket		
Rin	Input Impedance	16	kOhms
Fcut	-3dB down from 1kHz signal level at this frequency	13	Hz
Vin	Full scale input signal	1	Vrms
Vnoise	Equivalent input noise level, no signal applied	20	uVrms
THD	Total Harmonic Distortion @ 0.9Vrms Input	0.035	%
Notes	Test conditions as follows unless otherwise noted: 48kHz sample rate, test signal 1kHz sine wave and bandwidth measured is 20Hz to 20kHz		

**10.2. OPA2354**

## OPAx354 250-MHz, Rail-to-Rail I/O, CMOS Operational Amplifiers

### 1 Features

- Unity-Gain Bandwidth: 250 MHz
- Wide Bandwidth: 100-MHz GBW
- High Slew Rate: 150 V/ $\mu$ s
- Low Noise: 6.5 nV/ $\sqrt{\text{Hz}}$
- Rail-to-Rail I/O
- High Output Current: > 100 mA
- Excellent Video Performance:
  - Differential Gain: 0.02%, Differential Phase: 0.09°
  - 0.1-dB Gain Flatness: 40 MHz
- Low Input Bias Current: 3 pA
- Quiescent Current: 4.9 mA
- Thermal Shutdown
- Supply Range: 2.5 V to 5.5 V
- *Micro*SIZE and PowerPAD™ Packages

### 2 Applications

- Video Processing
- Ultrasound
- Optical Networking, Tunable Lasers
- Photodiode Transimpedance Amps
- Active Filters
- High-Speed Integrators
- Analog-to-Digital (A/D) Converter Input Buffers
- Digital-to-Analog (D/A) Converter Output Amplifiers
- Barcode Scanners
- Communications

### 3 Description

The OPAx354 series of high-speed, voltage-feedback CMOS operational amplifiers are designed for video and other applications requiring wide bandwidth. They are unity-gain stable and can drive large output currents. Differential gain is 0.02% and differential phase is 0.09°. Quiescent current is only 4.9 mA per channel.

The OPAx354 series of op amps are optimized for operation on single or dual supplies as low as 2.5 V ( $\pm 1.25$  V) and up to 5.5 V ( $\pm 2.75$  V). Common-mode input range extends beyond the supplies. The output swing is within 100 mV of the rails, supporting wide dynamic range.

For applications requiring the full 100-mA continuous output current, single and dual 8-pin HSOP PowerPAD versions are available.

The single version (OPA354) is available in the tiny 5-pin SOT-23 and 8-pin HSOP PowerPAD packages. The dual version (OPA2354) comes in the miniature 8-pin VSSOP and 8-pin HSOP PowerPAD packages. The quad version (OPA4354) is offered in 14-pin TSSOP and 14-pin SOIC packages.

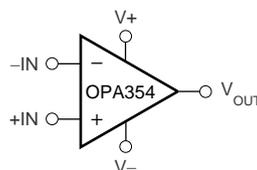
Multichannel version features completely independent circuitry for lowest crosstalk and freedom from interaction. All features are specified over the extended  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  temperature range.

#### Device Information<sup>(1)</sup>

PART NUMBER	PACKAGE	BODY SIZE (NOM)
OPA354	HSOP (8)	4.89 mm x 3.90 mm
	SOT-23 (5)	2.90 mm x 1.60 mm
OPA2354	VSSOP (8)	3.00 mm x 3.00 mm
	HSOP (8)	4.89 mm x 3.90 mm
OPA4354	SOIC (14)	8.65 mm x 3.91 mm
	TSSOP (14)	5.00 mm x 4.40 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

#### Simplified Schematic



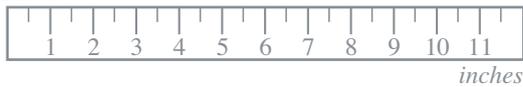
### **10.3. Hidrófono ITC-6050C**

# Model ITC-6050C

# Preamplified Hydrophone

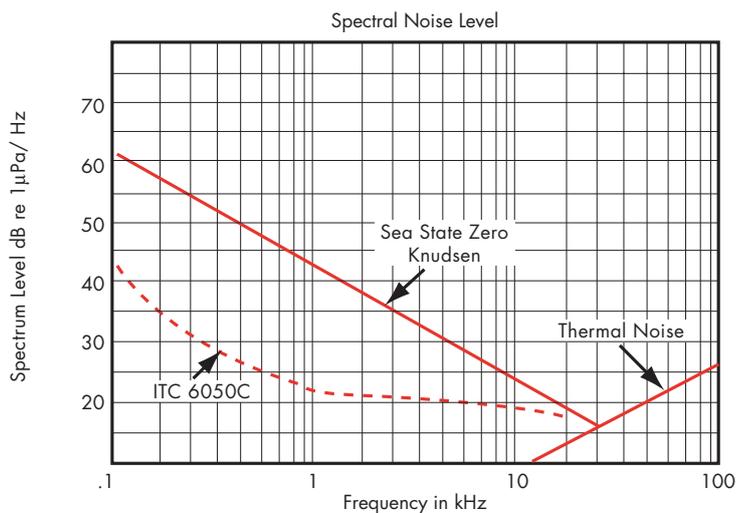
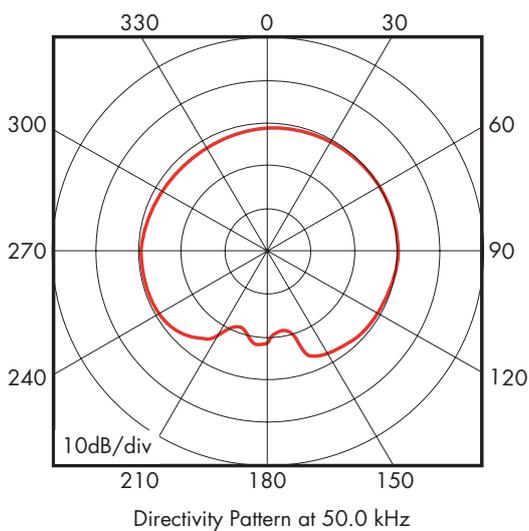
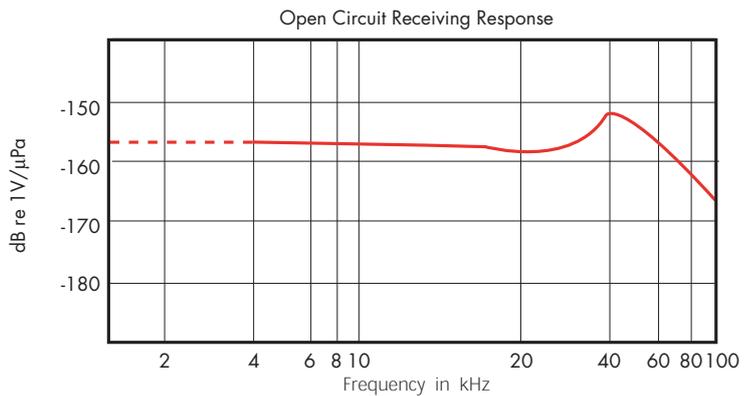
Model ITC-6050C

**The Model ITC-6050C** is a popular broadband hydrophone. The unit has a built-in, low noise preamplifier making it excellent for many field applications. Fully encapsulated in high quality polyurethane, it is well suited for rigorous conditions.



### Specifications (Nominal)

<b>Type</b>	Hydrophone w/ Preamplifier
<b>Resonance <math>f_r</math></b>	50 kHz
<b>Depth</b>	900 meters
<b>Envelope Dimensions (in.)</b>	2D x 12L
<b>Midband OCV</b>	-157 dB//1V/ $\mu$ Pa
<b>Suggested Band</b>	.03 - 70 kHz
<b>Beam Type</b>	Spherical



International Transducer Corporation

869 Ward Drive, Santa Barbara, CA 93111  
805.683.2575 • 805.967.8199 FAX

www.itc-transducers.com

## 10.4. Electret

**MODEL:** CMA-4544PF-W | **DESCRIPTION:** ELECTRET CONDENSER MICROPHONE

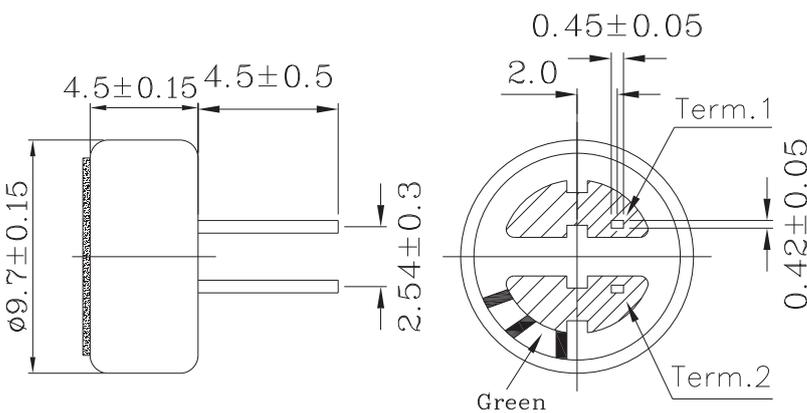
**SPECIFICATIONS**

parameter	conditions/description	min	typ	max	units
directivity	omnidirectional				
sensitivity (S)	f = 1 kHz, 1 Pa, 0 dB = 1 V/1 Pa	-46	-44	-42	dB
operating voltage			3	10	Vdc
output impedance (Zout)	f = 1 kHz, 1 Pa		2.2		KΩ
sensitivity reduction (ΔS-Vs)	f = 1 kHz, 1 Pa, Vs = 3.0 to 2.0 Vdc		-3		dB
frequency (f)		20		20,000	Hz
current consumption (IDSS)	Vs = 3.0 Vdc, RL = 2.2 KΩ			0.5	mA
signal to noise ratio (S/N)	f = 1 kHz, 1 Pa, A-weighted		60		dBA
operating temperature		-20		70	°C
storage temperature		-20		70	°C
dimension	ø9.7 x 4.5 mm				
weight				0.8	g
material	Al				
terminal	pin type (hand soldering only)				
RoHS	yes				

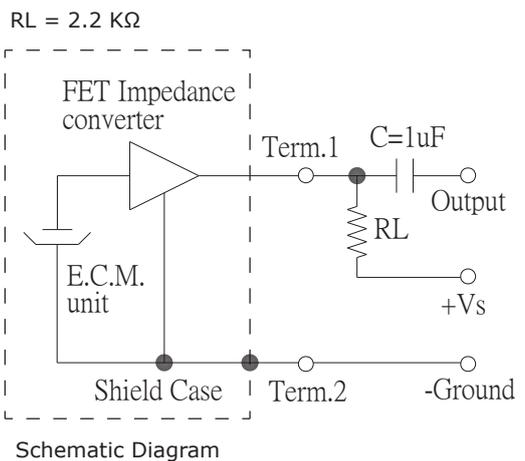
Note: We use the "Pascal (Pa)" indication of sensitivity as per the recommendation of I.E.C. (International Electrotechnical Commission). The sensitivity of "Pa" will increase 20dB compared to the "ubar" indication. Example: -60dB (0dB = 1V/ubar) = -40dB (1V/Pa)

**MECHANICAL DRAWING**

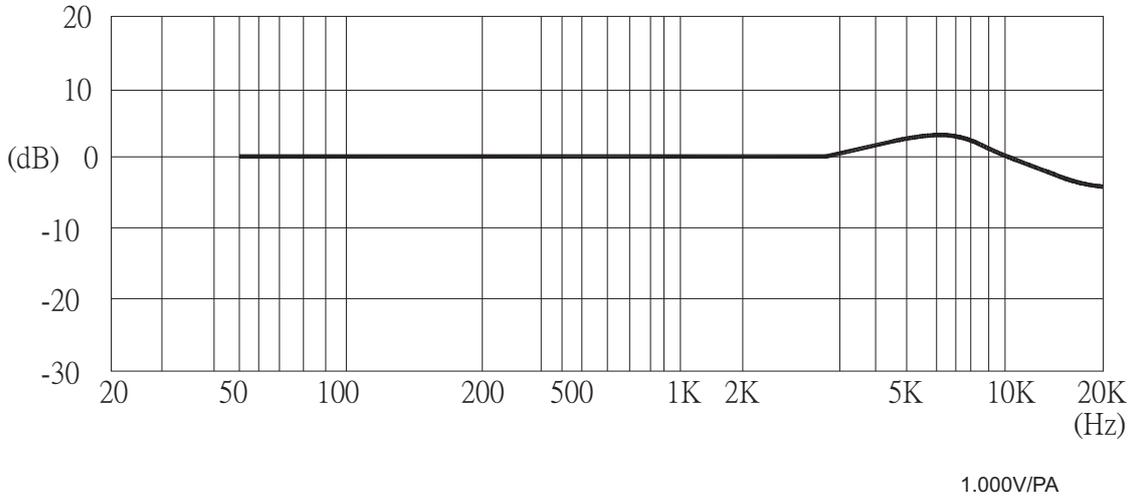
unit: mm



**MEASUREMENT CIRCUIT**



## FREQUENCY RESPONSE CURVE



## MECHANICAL CHARACTERISTICS

item	test condition	evaluation standard
soldering heat resistance	Soldering iron of $+270 \pm 5^\circ\text{C}$ should be placed on the terminal for $2 \pm 0.5$ seconds.	No interference in operation.
terminal mechanical strength	Apply to the terminal 4.9 N (0.5 kg) for 30 seconds	No damage or cutting off.
vibration test	The part should be measured after a vibration amplitude of 1.5 mm with 10~55 Hz band of vibration frequency to each of the 3 perpendicular directions for 2 hours.	After any tests, the sensitivity should be within $\pm 3$ dB of the initial sensitivity.
drop test	The part without packaging is subjected to 3 drops on each axis from the height of 1 m onto a 20 mm thick wooden board.	

## ENVIRONMENT TEST

item	test condition	evaluation standard
high temperature test	After being placed in a chamber at $+70^\circ\text{C}$ for 72 hours.	After any tests and 6 hours of conditioning at $+25^\circ\text{C}$ , the sensitivity should be within $\pm 3$ dB of the initial sensitivity.
low temperature test	After being placed in a chamber at $-20^\circ\text{C}$ for 72 hours.	
thermal shock	After being placed in a chamber at $+40^\circ\text{C}$ and 90 $\pm$ 5% RH for 240 hours.	
temperature cycle test	The part will be subjected to 10 cycles. One cycle will consist of:	

## TEST CONDITIONS

standard test conditions	a) Temperature: $+5 \sim +35^\circ\text{C}$	b) Humidity: 45 ~ 85%	c) Pressure: 860 ~ 1060 mbar
judgement test conditions	a) Temperature: $+25 \pm 2^\circ\text{C}$	b) Humidity: 60 ~ 70%	c) Pressure: 860 ~ 1060 mbar