



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Desarrollo de un módulo de efectos de sonido basado en
Rapsberry Pi para aplicaciones de DJ

Trabajo Fin de Grado

Grado en Ingeniería de Sistemas de Telecomunicación, Sonido e
Imagen

AUTOR/A: Pelfresne Pindado, Carlos

Tutor/a: Sanchis Rico, Juan Manuel

CURSO ACADÉMICO: 2022/2023

RESUMEN

En este trabajo se construye un módulo de efectos de sonido implementado en una Raspberry Pi para ser utilizado por un disc jockey en tiempo real. El módulo está basado en la aplicación hardware/software pi-Stomp desarrollada por Tree Fall Sound para efectos de guitarra controlados por pedalera. El trabajo mejora el diseño:

- Actualizando la Raspberry Pi a un modelo más actual y por tanto más potente con el fin de reducir el tiempo de procesamiento y el retardo asociado, permitiendo la conexión del módulo de efectos por envío y retorno de auxiliares o por inserto.
- Ofreciendo un interfaz de usuario amplia que permite el control de los parámetros de cada efecto incluyendo pedalera, controles rotatorios, pantalla...
- Montando el sistema en una caja de sobremesa robusta y de fácil manejo para el DJ.

Palabras clave: Raspberry Pi; módulo de efectos sonoros; Disk Jockey.

ABSTRACT

In this work we intend to build a sound effects module implemented in a Raspberry Pi to be used by a disc jockey in real time. The module will be based on the hardware/software application pi-Stomp developed by Tree Fall Sound for pedalboard-controlled guitar effects. The work aims to improve the design, setting the following objectives:

- Upgrade the Raspberry pi to a more current model and therefore more powerful to reduce the processing time and the associated delay, allowing the connection of the effects module by send and return of auxiliaries or by insert.
- Offer a wide user interface that allows the control of the parameters of each effect including pedalboard, rotary controls, display...
- Assemble the system in a robust desktop box that is easy to use for the DJ.

Key words: Raspberry Pi; sound effects module; Disk Jockey.

INDICE

1. INTRODUCCIÓN.....	4
1.1 Motivación.....	4
1.2 Objetivos	4
1.3 Metodología y planificación.....	5
1.3 Estructura de la memoria.....	6
2. ESTADO DEL ARTE – INTRODUCCION TEORICA	7
2.1 Equipos existentes en el mercado	7
3. KIT DE TREEFALLSOUND	9
3.1 Diseño original del proyecto	10
3.2 Arquitectura y funcionamiento del software	11
3.3 Modificaciones para adaptación a aplicaciones de DJ	12
4. PLAN DE DESARROLLO	14
4.1 Diseño de hardware	14
4.2 Ensamblaje del hardware	15
4.2.1 Soldadura de componentes electrónicos en PCB de pi-Stomp	15
4.2.2 Montaje del conjunto de placas PCB	18
4.2.3 Conexionado de los controles analógicos	19
4.3 Instalación del software	21
4.3.1 Interfaz gráfica de piRMX en pantalla LCD	22
4.3.2 Configuración de controles analógicos externos	23
5. PRUEBAS FUNCIONALES	25
5.1 Prueba en el laboratorio de instrumentación	25
5.1.1 Asignación de controles analógicos en MOD-GUI.....	25
5.2 Prueba en un entorno real con una mesa de mezclas para DJ.....	26
5.2.1 Problema de ruido al procesar señales de entrada	28
6. MONTAJE DENTRO DE UNA CAJA METÁLICA.....	31
6.1 Primer boceto del diseño	31
6.2 Problema de incompatibilidad en los conectores del panel trasero	32
6.3 Maquetación y montaje en la caja	34
7. CONCLUSIONES Y POSIBLES MEJORAS	36
7.1 Trabajos futuros.....	36
7.1.1 Detección automática de BPM.....	36
7.1.2 Personalización de la interfaz gráfica y pantalla táctil	37
7.1.3 Añadir más controles analógicos	37

8. BIBLIOGRAFÍA	38
9. ANEXOS	40
Anexo I. Relación del trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030. ...	40
Anexo II. Diccionario de acrónimos.....	42
Anexo III. Coste.....	43
Anexo IV. Fotos montaje final de piRMX.....	44

1. INTRODUCCIÓN

1.1 Motivación

Desde pequeño me ha gustado la música electrónica, siempre la he escuchado, hasta que un día tuve la oportunidad de subirme a la cabina de un DJ y me quedé embobado viendo cómo conseguía controlar la música a través de un montón de botones y controles que no entendía, entonces me picó la curiosidad. Empecé a buscar información por internet, mirar páginas web de los fabricantes, tutoriales de cómo sincronizar canciones, aplicar efectos, etc... Por fin pude comprarme una controladora DJ y dediqué horas y horas a mezclar canciones, entender cada botón y exprimir todas sus funcionalidades. Hasta que un día dejó de funcionar, mi afán friki y la curiosidad me llevaron a desmontarla pieza a pieza y arreglarla por mi cuenta. Desde ahí tuve claro que quería estudiar algo relacionado con la electrónica y el sonido.

Hoy en día sigo teniendo esa misma mesa de mezclas, pero llevo tiempo echando en falta tener más controles sobre los efectos a la hora de mezclar canciones. Los módulos de efectos que existen en el mercado son muy caros, por eso me puse a investigar si había algunas alternativas, hasta que encontré varios proyectos en Internet de usuarios que había construido su propio módulo de efectos. Di con el proyecto de <https://www.treefallsound.com/>, una pedalera de efectos para guitarristas/bajistas con infinidad de efectos y una interfaz gráfica muy completa e intuitiva, además de contar con un software de código abierto. Mi idea era llevar a cabo el proyecto original para luego poder modificarlo cumpliendo con mis requerimientos personales a la hora de mezclar.

Pienso que el TFG es una oportunidad perfecta para poder ampliar mis conocimientos sobre electrónica aplicada al audio y aprender sobre el diseño conjunto de hardware y software para conseguir un módulo de efectos de sonido personal, profesional y funcional para aplicaciones de DJ.

1.2 Objetivos

En este trabajo se pretende construir un módulo de efectos de sonido implementado en una Raspberry pi para ser utilizado por un disc jockey en tiempo real.

El módulo estará basado en la aplicación hardware/software pi-Stomp desarrollada por Tree Fall Sound para efectos de guitarra controlados por pedalera.

El trabajo pretende mejorar el diseño, fijando los siguientes objetivos secundarios:

- actualizar la Raspberry pi a un modelo más actual y por tanto más potente con el fin de reducir el tiempo de procesado y el retardo asociado, permitiendo la conexión del módulo de efectos por envío y retorno de auxiliares o por inserto;

- ofrecer un interfaz de usuario amplia que permita el control de los parámetros de cada efecto incluyendo pedalera, controles rotatorios, display...
- actualizar el software, incluyendo un algoritmo de detección del BPM, que permita la aplicación de efectos síncronos con el tempo de la pieza musical
- montar el sistema en una caja de sobremesa robusta y de fácil manejo para el DJ.

1.3 Metodología y planificación

Primero realicé un trabajo de investigación en Internet, busqué varios proyectos que buscaban dar solución al problema propuesto, construir un módulo de efectos de sonido desde cero a partir de hardware y software personalizable y escalable . Finalmente me decanté por el proyecto de Tree Fall Sound ya que estaba muy bien documentado y además pude hablar con el creador para preguntarle algunas dudas que me surgieron a la hora de comprar los componentes hardware y sobre algunas modificaciones del software.

Teniendo en cuenta mi experiencia con los equipos de DJ, las posibilidades del proyecto e investigando en internet me puse a pensar en el diseño de los efectos y los controles que necesitaba para poder montar un hardware funcional, intuitivo y personal.

Una vez tenía en mente la idea general del proyecto me dispuse a comprar todos los componentes de hardware necesarios. Tuve algunos problemas para encontrar todos los componentes ya que en el momento había una falta de suministros a nivel mundial y tuve que dedicar dos semanas hasta que conseguí comprar todo el material a través de Internet.

Cuando tenía ya en mis manos todos los componentes me dispuse a soldar todos los componentes en la placa PCB que haría de interfaz entre los diferentes componentes siguiendo la documentación del proyecto. Con la PCB principal de Tree Fall Sound operativa ya pude conectar el resto de las placas PCB como la tarjeta de sonido, el HatHackker, y la Raspberry Pi para completar el montaje básico del hardware.

Después instalé todo el software necesario en la RPi e hice dos pruebas funcionales, una en el laboratorio de instrumentación de la EPSG y otra en mi piso implementado el piRMX con otros equipos de DJ.

Tras solucionar algunos problemas con el hardware y comprobar que el piRMX cumplía con los requerimientos de funcionalidad y calidad tuve que montar todo el conjunto dentro de una caja metálica para que fuese práctico, portátil y profesional. Hice un diseño más elaborado tomando medidas reales del conjunto para elegir la caja más apropiada e hice los correspondientes cortes y agujeros en la misma.

Debido a que es mi primer proyecto relacionado con electrónica y audio he tenido que hacer frente a varios problemas que he ido solucionando por lo que no he tenido tiempo de trabajar en las mejoras de software a pesar de tenerlas presentes durante todo el proyecto.

1.3 Estructura de la memoria

Este trabajo se estructura en varios puntos en los que se tratan los siguientes temas:

- Breve introducción a los módulos de efectos para aplicaciones de DJ.
- Elección de los componentes y hardware a partir del proyecto de Tree Fall Sound
- Diseño de la disposición de los componentes como botones y potenciómetros a partir de los requerimientos personales
- Proceso de ensamblado y montaje del kit
- Pruebas funcionales del conjunto hardware-software
- Diseño y montaje del hardware dentro de una caja metálica pensando en su uso dentro de una cabina de DJ
- Conclusiones y trabajos futuros.

2. ESTADO DEL ARTE – INTRODUCCION TEORICA

La figura del DJ (*disk-jockey*) empezó a aparecer hacia finales de los años 1950 en los guetos de Kingston (Jamaica), donde los promotores organizaban grandes fiestas en las calles que estaban centradas en torno a la actividad del *disk-jockey*, que tocaba música mediante dos tocadiscos utilizando grandes sistemas de amplificación de sonido [1].

Hacia mediados de los años 1960 los clubs nocturnos y discotecas seguían creciendo en Europa y Estados Unidos. Empezaron a aparecer tocadiscos y mesas de mezclas específicos para DJs. En 1969 el DJ Francis Grasso popularizó la técnica del *beat-matching*, que consistía en crear transiciones imperceptibles entre vinilos mediante la sincronización de los tiempos de ambos.

Con la aparición de los formatos digitales para la distribución de música, los fabricantes de equipos de sonido empezaron a incorporar nuevos botones y funcionalidades en los reproductores de CDs. Con el formato digital surgían numerosas ideas para potenciar la creatividad en directo del DJ, como podían ser *loops* (bucles), puntos de *cue* (escucha o monitorización), ajustes más finos en la velocidad de reproducción, etc. En definitiva, los equipos facilitaban enormemente las tareas de sincronización entre canciones hasta tal punto que hoy en día algunos consiguen realizar las mezclas de forma automática a través del análisis frecuencial de las canciones.

Estos avances permiten al artista poder centrarse más en la improvisación del momento, mezclando más de dos canciones a la vez, añadiendo cajas de ritmos, aplicando efectos, editando bucles, y todo esto en directo. Los DJs modernos ya no pinchan canciones, sino que crean música en directo [2].

2.1 Equipos existentes en el mercado

Los módulos de efectos portátiles más populares son probablemente los efectos de pedal que se pueden ver generalmente a los pies de guitarristas o bajistas. Suelen tener varios de ellos al alcance de sus pies para poder activarlos o desactivarlos en directo. Cada pedal es un efecto de sonido por lo que se conectan en cascada para formar una pedalera de efectos, cambiando la ruta de conexionado podemos conseguir diferentes efectos complejos de sonido (Ilustración 1).

Esta ha sido la forma tradicional de aplicar efectos para músicos en directo, para los DJs no es práctico utilizar pedales ya que todo su trabajo se desarrolla sobre una mesa y manipula los dispositivos con las manos. Como el DJ tiene en sus manos una mesa de mezclas suele utilizar los efectos en una configuración de envío y retorno para tener un mayor control sobre las señales de audio a las que desea aplicar el efecto, además de poder controlar los diferentes parámetros de cada efecto como puede ser el tempo, resonancia, *decay*, tono... Las marcas de equipos para DJs han sacado al mercado varias soluciones que incorporan múltiples efectos digitales en un único aparato, la mayoría de ellos son caros y permiten aplicar un solo

efecto a la vez con varios parámetros configurables que cambian en función del efecto seleccionado. El estándar en la mayoría de las discotecas y festivales es el Pioneer RMX-1000 (Ilustración 2), y su precio ronda los 700€.

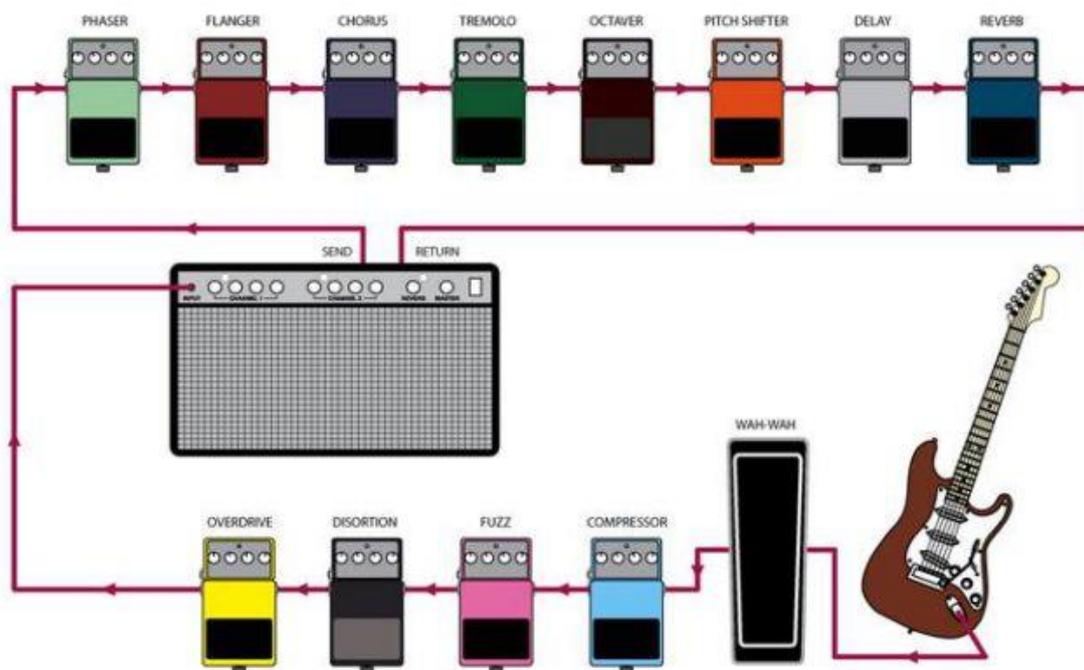


Ilustración 1. Cadena de efectos de guitarra. Fuente: clasesguitarraonline.com



Ilustración 2. Módulo de efectos Pioneer RMX-1000. Fuente: pioneerdj.com

Existen otras soluciones más asequibles de otros fabricantes como es el caso del Korg Mini Kaoss Pad (Ilustración 3), que ronda los 150€. Dispone de muchos menos botones y funcionalidades, pero la forma de aplicar los efectos es muy expresiva a través de un panel táctil X-Y.



Ilustración 3. Módulo de Efectos Korg Mini Kaoss Pad. Fuente: korg.com

3. KIT DE TREEFALLSOUND

Como bien he comentado anteriormente, mi idea es adaptar el proyecto de Tree Fall Sound. Se trata de un módulo de efectos de alta calidad (Ilustración 4) para guitarra, bajo o teclado que se puede personalizar gracias a la documentación disponible en su página web y al software de código abierto. Tiene más de 300 efectos digitales basados en pedales de guitarra que se pueden configurar y editar desde un ordenador o Tablet, accediendo a la interfaz gráfica de MOD (<https://mod.audio/platform/>). MOD también vende su propio hardware, pero está diseñado como pedalera para guitarristas y sólo dispone de 3 botones y 3 potenciómetros, su precio es de 500€.



Ilustración 4. pi-Stomp Core v2.0.

3.2 Arquitectura y funcionamiento del software

El software pi-Stomp se basa en un sistema operativo Raspberry Pi de 64 bits y utiliza MOD [4], una plataforma de software de código abierto creada por moddevices.com, que incluye **MOD UI** y **MOD Host**. MOD UI es una interfaz de usuario basada en web para la creación y edición de pedaleras virtuales, mientras que MOD Host es el motor de audio que aloja y procesa los complementos LV2 github.com.

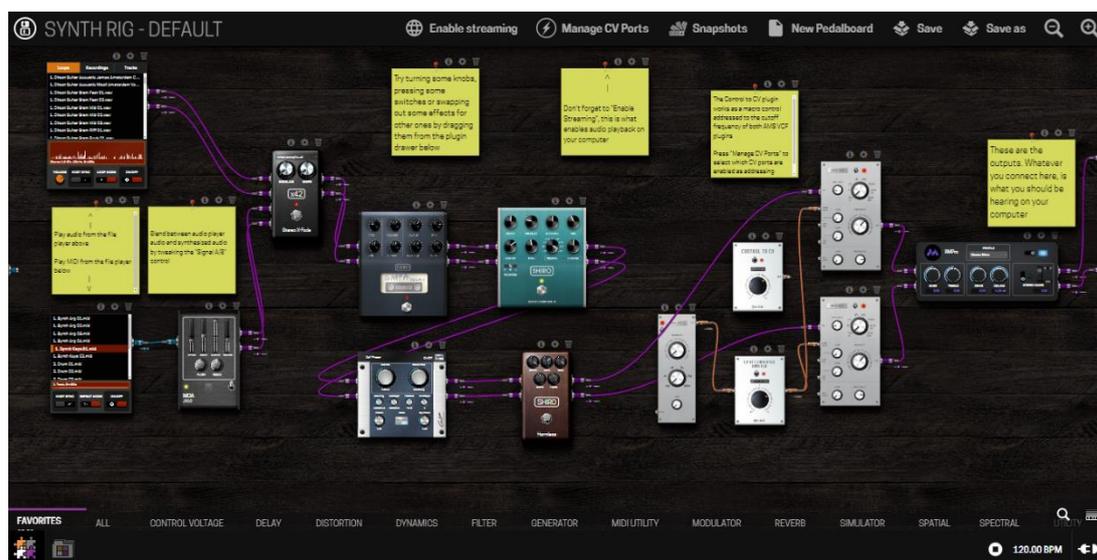


Ilustración 6. MOD Audio GUI.

El proyecto pi-Stomp también incluye un servicio de software llamado **mod-ala-pi-stomp**, que se encarga de interactuar con el hardware de pi-Stomp (LCD, potenciómetros, codificadores, interruptores de pie, MIDI, etc.). Este servicio utiliza controladores para monitorear todos los dispositivos de entrada, controlar la pantalla LCD y enviar comandos a mod-host para leer y escribir información de configuración de la pedalera, entre otras funciones (github.com).

El servicio **mod-ala-pi-stomp** se compone de dos componentes principales: **modalapistomp.py** y el módulo **modalapi**. Estos componentes trabajan juntos para proporcionar la funcionalidad necesaria para interactuar con el hardware y el software MOD (github.com).

El repositorio de pi-Stomp en GitHub también incluye *scripts* de configuración que automatizan la descarga e instalación de todas las dependencias necesarias, como:

- Dependencias de Python
- Software MOD (MOD UI y MOD Host)
- Controladores de tarjeta de sonido
- Ajustes del sistema
- Cientos de complementos LV2
- Pedaleras de muestra

3.3 Modificaciones para adaptación a aplicaciones de DJ

A la hora de adaptar el proyecto para ser utilizado junto a una mesa de mezclas y conectar el módulo de efectos mediante un envío y retorno, como el ejemplo mostrado en Ilustración 7, lo primero que pensé fue en la posible latencia que podía haber entre la entrada y la salida del módulo de efectos.

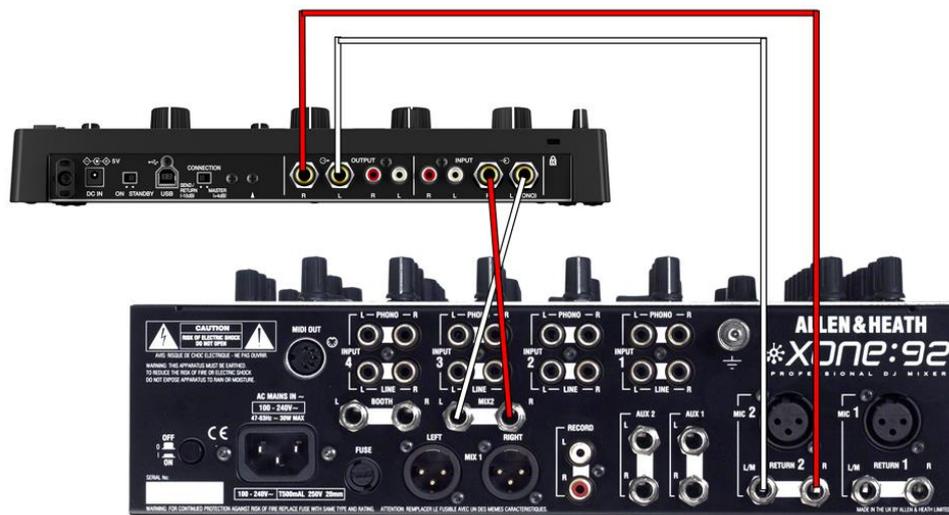


Ilustración 7. Conexión de un módulo de efectos en bucle envío/retorno.

Si el módulo de efectos tiene mucha latencia podría ser molesto a la hora de mezclar diferentes señales e incluso sonar mal. Por suerte pude hablar con el creador del proyecto y me redireccionó a una página web en la cual había hecho un pequeño estudio comparando las latencias en función del rendimiento de los conversores de audio, es decir cambiando la frecuencia de muestreo y el tamaño del buffer. Utilizando una frecuencia de muestreo de 48 kHz y un tamaño de buffer de 128 muestras la latencia entre la entrada y la salida del módulo de efectos es de 5,8 milisegundos. Según algunos estudios el cerebro humano empieza a sentir los efectos de una señal retardada a partir de los 12-15 milisegundos, por lo que con esta configuración no deberíamos de tener ningún tipo de problema [5].

Quiero poder expresar al máximo las funcionalidades de múltiples efectos a la vez, y esto podría comprometer el rendimiento de la CPU, así que voy a utilizar una Raspberry Pi 4B que tiene una mayor velocidad de procesamiento, entorno a un 8% más que la RPi 3A+, y una mayor capacidad de RAM, pasando de 512MB a 4 GB. Cambiar de Raspberry Pi será un quebradero de cabeza a la hora de montar todo el hardware dentro de una caja ya que la RPi 4 es 2cm más larga por el lateral donde estarán los conectores de audio y de alimentación de la placa PCB de TreeFallSound, así que habrá que pensar en alguna solución con unos *jumpers* o alargadores para las conexiones de alimentación y de audio del módulo.

El diseño original del pi-Stomp está hecho con tres botones para pie y dos resistencias variables para controlar parámetros de los efectos (Ilustración 4), en este caso no tiene ningún sentido utilizar botones de pie así que voy a adaptar el diseño de los componentes para mis

necesidades personales. Prefiero tener tan sólo la capacidad de controlar dos efectos, pero tener más controles sobre los parámetros de cada uno de ellos, por supuesto estos dependerán del efecto seleccionado. El diseño de la distribución de los componentes se hará más adelante.

Por último, he decidido utilizar una pantalla LCD un poquito más grande, en vez de 2,2" he decidido optar por una pantalla de 2,8" compatible que además tiene detección táctil, en un futuro sería interesante poder implementar una interfaz gráfica diferente, incluyendo la navegación táctil o incluso la posibilidad de utilizar el panel táctil como un eje XY para modificar a la vez dos parámetros de un efecto.

4. PLAN DE DESARROLLO

Dada la dimensión del proyecto y mi inexperiencia tanto en el ámbito de la electrónica como en el ámbito de sistemas Linux y programación, es de vital importancia tener muy claro los pasos a seguir para poder llevar a cabo todos los objetivos. Con el planteamiento inicial hecho, teniendo en cuenta el proyecto del que partimos y con las modificaciones que quiero implementar, el primer paso a dar es conseguir un diseño de la disposición de todos los controles y componentes para después poder ensamblar todo el hardware, y finalmente poder instalar el software original de pi-Stomp para probar el correcto funcionamiento de la base del proyecto.

4.1 Diseño de hardware

La placa PCB de pi-Stomp cuenta con un encoder que se utiliza para la navegación a través de los diferentes menús de configuración, también cuenta con un potenciómetro para controlar directamente el volumen del nivel de salida del módulo de efectos.

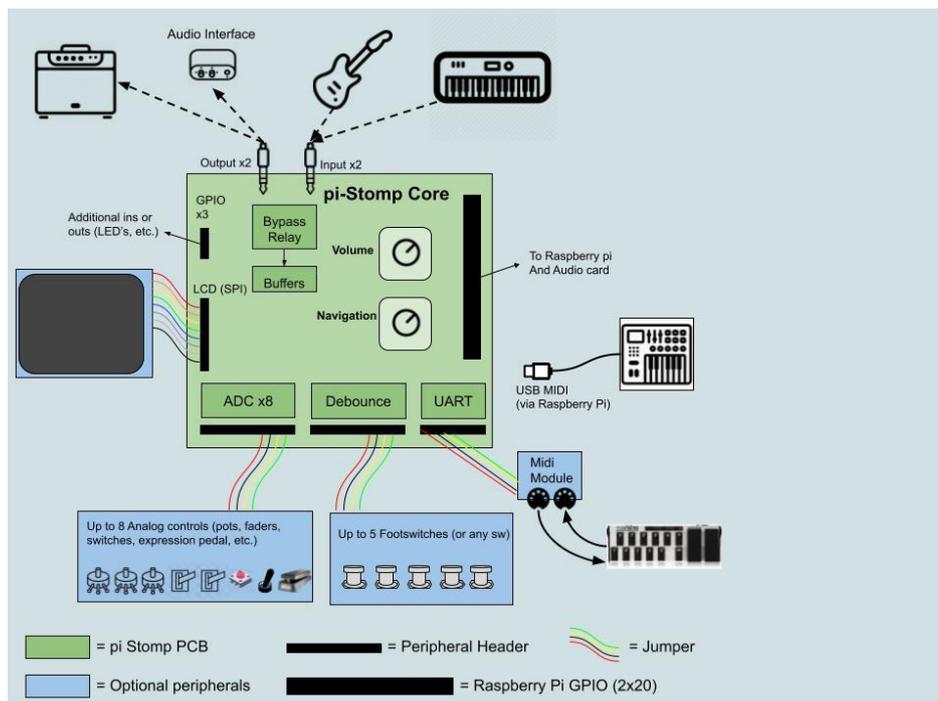


Ilustración 8. Arquitectura e interfaz del hardware de pi-Stomp.

Teniendo en cuenta las posibilidades de conexiones de hardware del proyecto de pi-Stomp (Ilustración 8) y tomando como referencia algunos de los equipos existentes en el mercado comentados anteriormente he decidido que mi módulo Pi de efectos cuente con los siguientes controles en el panel frontal:

- EFECTO 1: 1 botón ON/OFF, 1 potenciómetro DRY/WET y 2 potenciómetros para controlar otros parámetros del efecto.

- EFECTO 2: 1 botón ON/OFF, 1 potenciómetro DRY/WET y 2 potenciómetros para controlar otros parámetros del efecto.

Sería interesante tener algún botón o *encoder* para poder controlar o modificar el BPM de los efectos. En principio el software debería ser capaz de analizar el BPM del sonido en directo, pero no he podido realizar la prueba con la interfaz de MOD audio gratuita hasta que todo el hardware y software funcionen en conjunto, así que más tarde veremos si es necesario implementar esta funcionalidad. En un futuro me gustaría poder visualizar el BPM en la pantalla LCD del módulo para tener una referencia visual rápida sin necesidad de mirar la interfaz gráfica de MOD en un ordenador o Tablet.

4.2 Ensamblaje del hardware

Una vez tuve todos los componentes que adquirí a través de Randall, el creador de pi-Stomp, y otras tiendas online de componentes electrónicos me dispuse a comprobar que contaba con todos los componentes con la ayuda de la lista de materiales disponible en la página web del proyecto.

4.2.1 Soldadura de componentes electrónicos en PCB de pi-Stomp

Siguiendo las instrucciones del proyecto me dispuse a soldar todos los componentes y conectores en la placa PCB de pi-Stomp. Esta labor la desarrollé en el laboratorio de instrumentación de la EPSG (Ilustración 9).

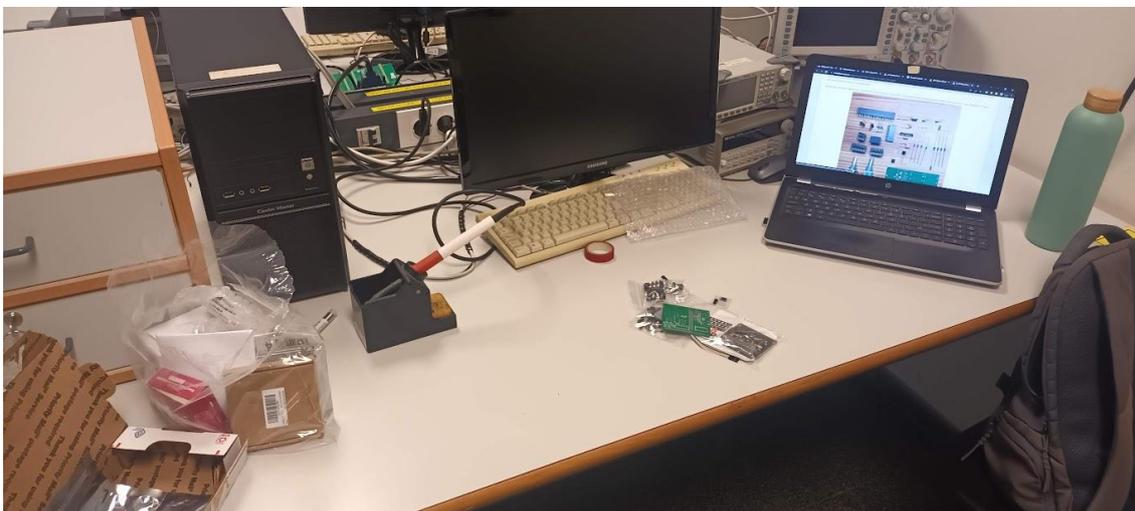


Ilustración 9. Montaje para soldar en un laboratorio de la EPSG (UPV).

Tenía poca experiencia soldando así que tuve mucho cuidado para no sobrecalentar los componentes ni provocar ningún cortocircuito entre pines ya que algunos de ellos estaban muy juntos. Como se puede ver en la Ilustración 5, la placa PCB no tenía soldado ningún componente así que empecé por soldar componentes simples como resistencias, condensadores y transistores (Ilustración 10).

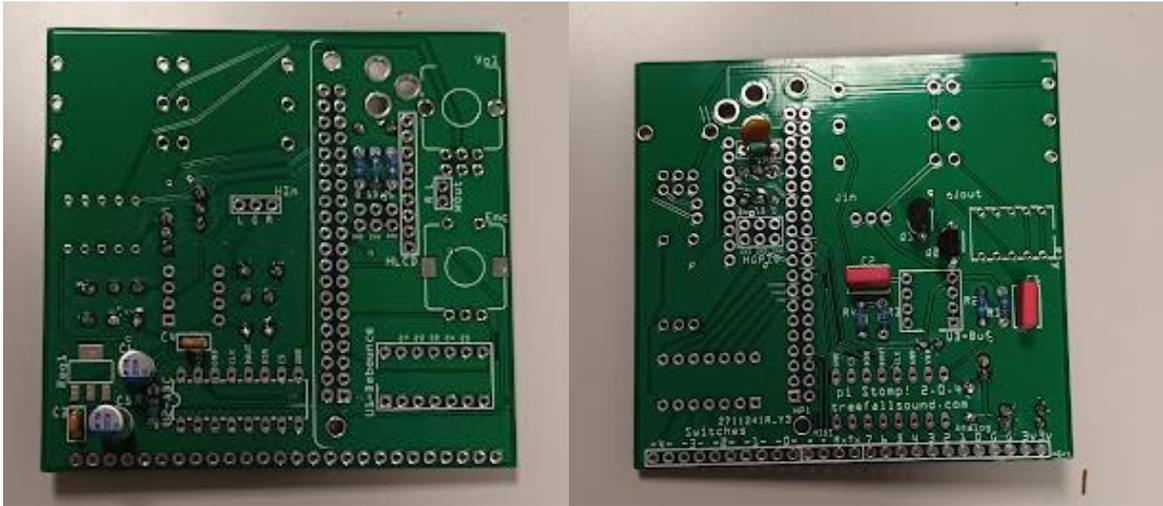


Ilustración 10. Montaje de componentes básicos en la PCB de pi-Stomp, parte trasera (izq.) y frontal (der.)

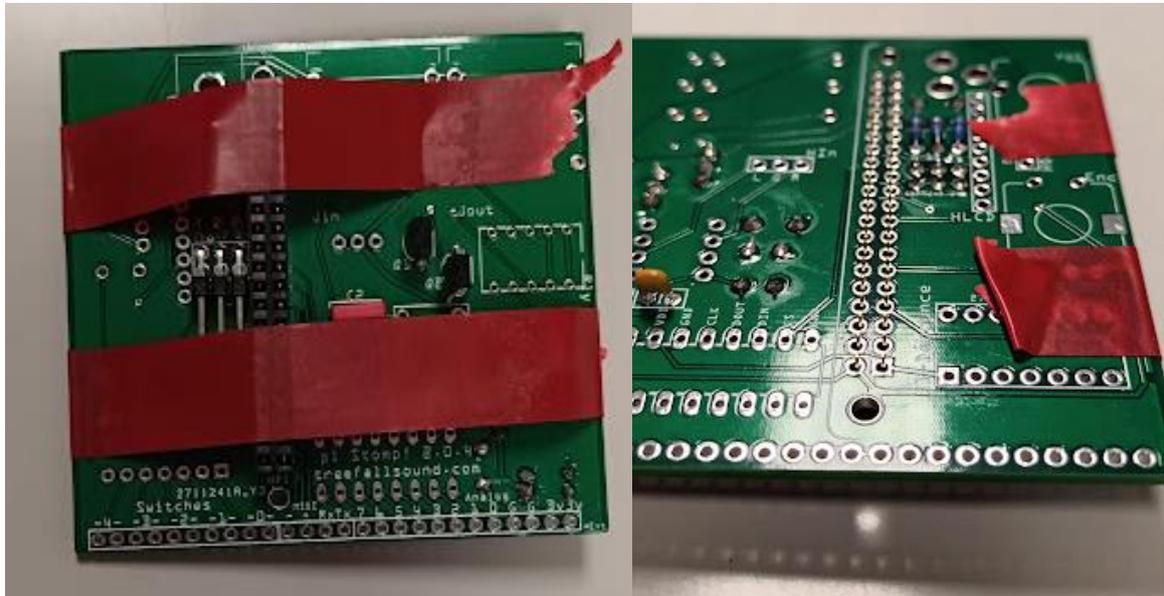


Ilustración 11. Montaje de componentes en PCB.

Para soldar los *sockets*, *headers*, conectores, potenciómetros y demás componentes más complicados me tuve que ayudar con un poco de cinta aislante para aguantarlos en su posición mientras aplicaba la soldadura (Ilustración 11).

Después coloqué cada chip en su correspondiente socket y conecté los cables de la entrada de audio en la PCB (Ilustración 12), esto se debe hacer antes de colocar el LCD ya que después no se tiene acceso a este conector.

El siguiente paso era soldar la pantalla LCD a la placa base (Ilustración 13). Pero al tratarse de una pantalla táctil contaba con 14 pines que no permitían encajar la pantalla en la placa que tiene 9 entradas para la misma. Como bien he comentado anteriormente, en el futuro me gustaría poder cambiar la interfaz gráfica e implementar algunas funcionalidades de navegación a través de la pantalla táctil, por lo que opté por quitar los 5 pines que se usan para la comunicación a través de la interfaz SPI y soldar directamente unos cables en esos

pinos que sobraban, de esta manera ya pude soldar la pantalla y mantener el acceso a los pines del panel táctil. También apliqué un poquito de pegamento caliente entre la pantalla y la PCB de pi-Stomp para conseguir una mayor resistencia mecánica y evitar corto circuitos ya que tenía algunas partes metálicas expuestas.



Ilustración 12. Placa PCB con chips y potenciómetros.

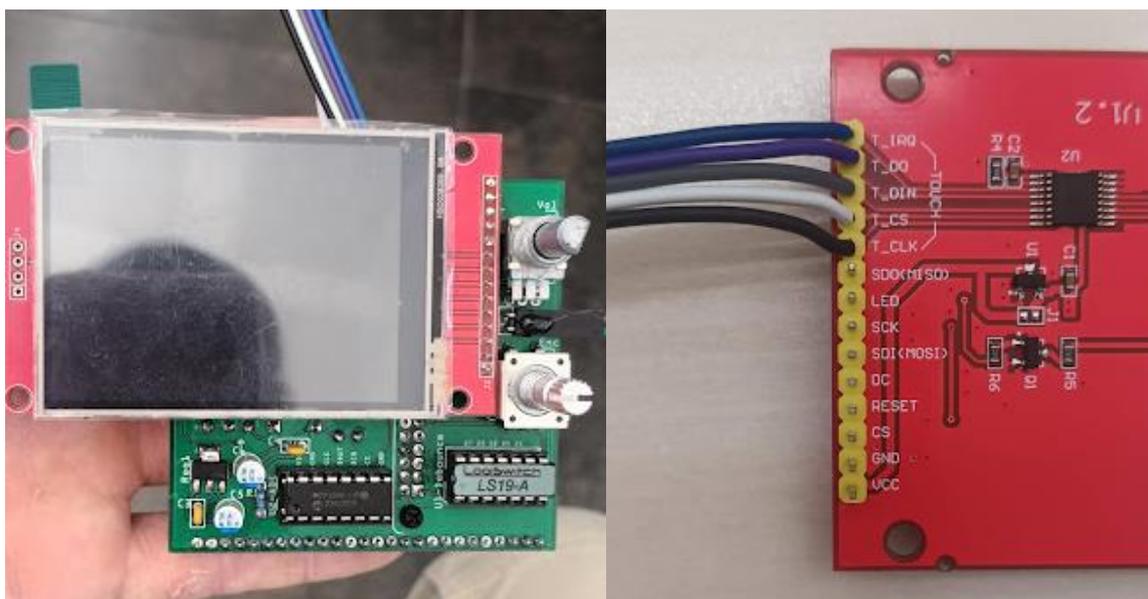


Ilustración 13. Colocación de la pantalla LCD PCB panel táctil en la placa PCB con jumper.

4.2.2 Montaje del conjunto de placas PCB

Con la PCB de pi-Stomp lista con todos sus componentes llegó el momento de encajar todas las piezas que conforman el hardware, para ello se utilizan unos pequeños separadores de plástico y unos tornillos. En la Ilustración 14 se muestran de izquierda a derecha las diferentes placas: PCB de pi-Stomp (sin el LCD todavía), Hatt-Hacker para facilitar y duplicar las conexiones del GPIO de la RPI, tarjeta de sonido IQ-Audio, y finalmente la Raspberry Pi 4B.

En este punto pude confirmar el problema a la hora de diseñar una caja para el conjunto ya que la RPi4 es más larga que la RPi3, al montar las placas unas encima de otras se aprecian como los conectores de entrada y salida de audio y alimentación de la PCB de pi-Stomp quedan más profundos que los de puertos de la RPi4. Simplemente coloqué cada placa en su posición sin encajar los pines de conexión y ya se puede apreciar en la Ilustración 15.

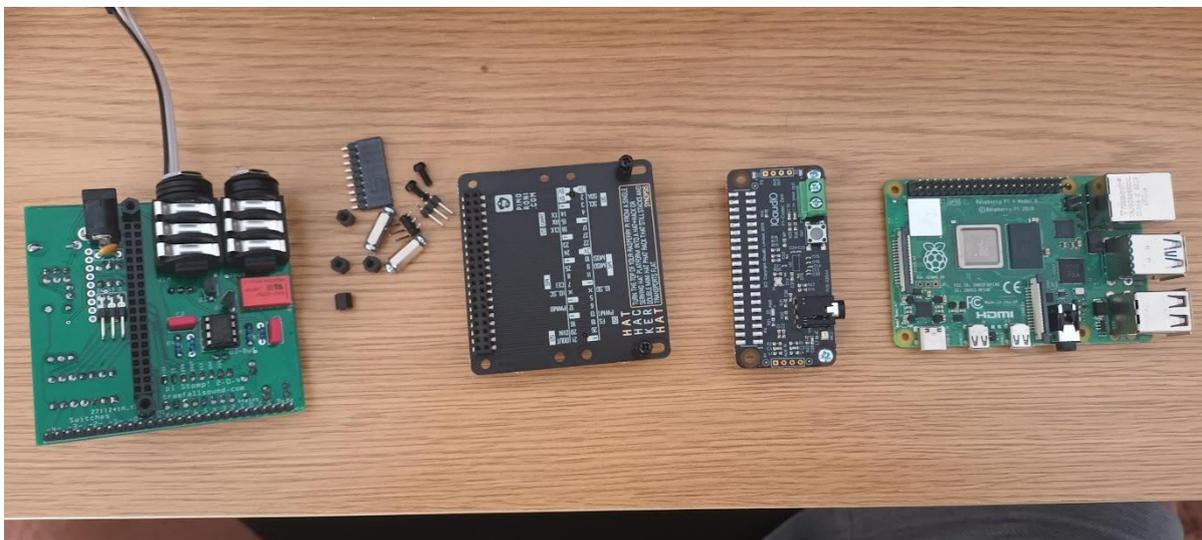


Ilustración 14. Diferentes placas que conforman el proyecto

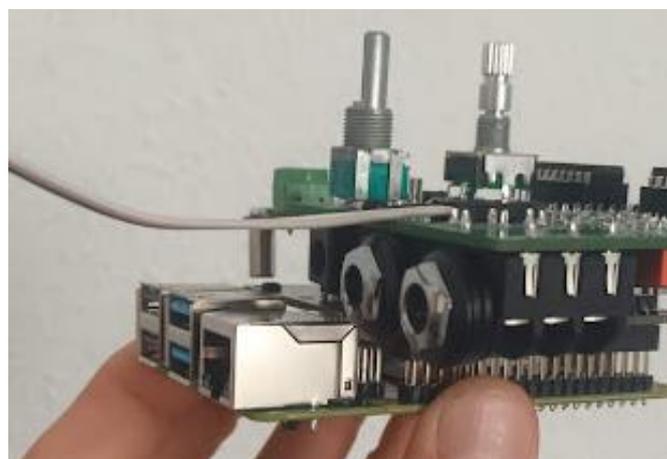


Ilustración 15. Presentación de montaje conjunto de hardware

Es un gran problema ya que mi idea es montar todo el hardware en una caja robusta pero con esta diferencia entre las profundidades de los conectores es imposible, las únicas dos soluciones que se me ocurren son hacer una caja a ras de los conectores de audio y alimentación, y que entonces los puertos de la RPi4 y parte de la placa queden por fuera, lo cual es una chapuza y estéticamente no queda nada bonito ni es práctico a la hora de transportarlo; la otra opción es hacer la caja a ras de los conectores de la RPi4 y desoldar los dos conectores de audio y el de alimentación para colocarlos de alguna manera en la caja y conectarlos a la placa mediante unos pequeños cables o *jumpers*, lo cual es más estético y profesional. He decidido omitir este problema por el momento para seguir progresando con el proyecto y ya lo solucionaré a la hora de diseñar la caja.

El siguiente problema de montaje no tardó en aparecer, después de montar y encajar unas placas encima de otras, me dispuse a conectar los cables de audio pero los tres cables para la entrada de audio en la placa de pi-Stomp no llegaban hasta la tarjeta de sonido, y no podía cambiarlos por otros más largos porque estaban soldados entre la placa de pi-Stomp y la pantalla LCD. La solución más viable era cortar los jumpers y soldarles otros cables para alargarlos un par de centímetros (Ilustración 16).

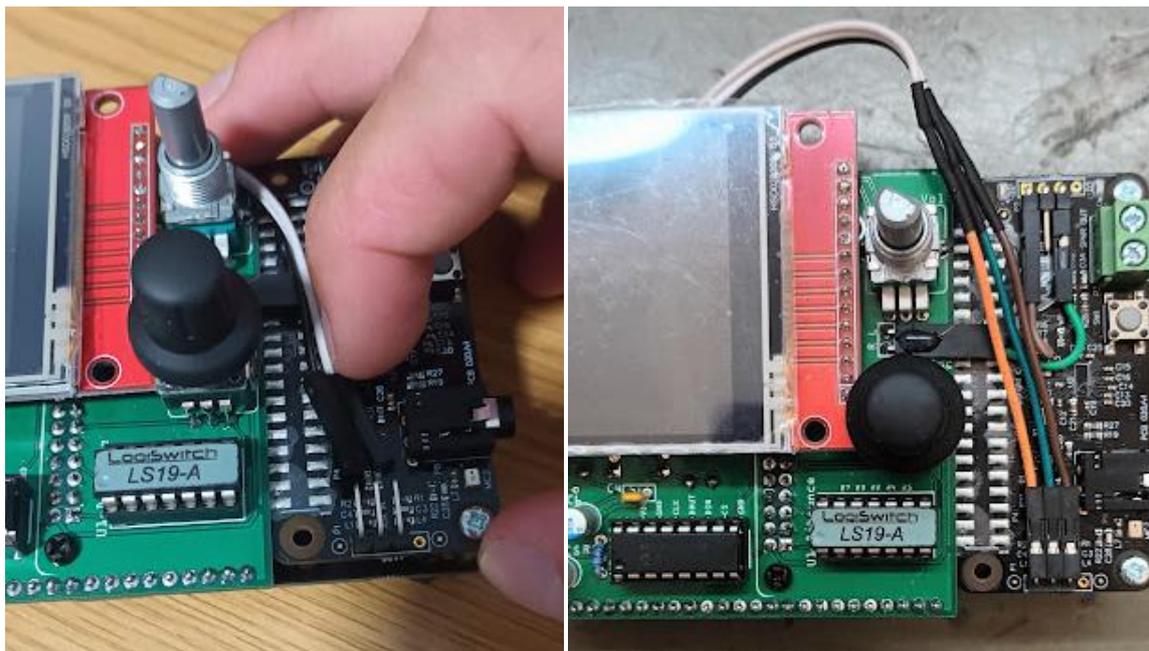


Ilustración 16. Conexiones de la tarjeta de sonido antes y después de soldar cables alargadores.

4.2.3 Conexión de los controles analógicos

Ahora con todas las placas apiladas se pueden conectar los potenciómetros y botones correspondientes. Primero se puede probar a conectar los componentes externos en una protoboard para que sea más fácil cambiar la configuración y las conexiones de los componentes en caso de fallo. En la Ilustración 17 se puede observar el diagrama de conexión de los controles analógicos que he decidido utilizar en el proyecto haciendo uso de los pines disponibles en la PCB del piRMX. A la izquierda se encuentran los dos botones S1 y S2, cada uno tiene un LED para indicar si dicho efecto está activado o no, para los LEDs se

utilizan otros pines dedicados para este propósito que se encuentran escondidos tras la pantalla LCD. A la derecha se encuentra el diagrama de conexionado de los potenciómetros rotatorios, cada uno de ellos con un valor resistivo de 10 k Ω .

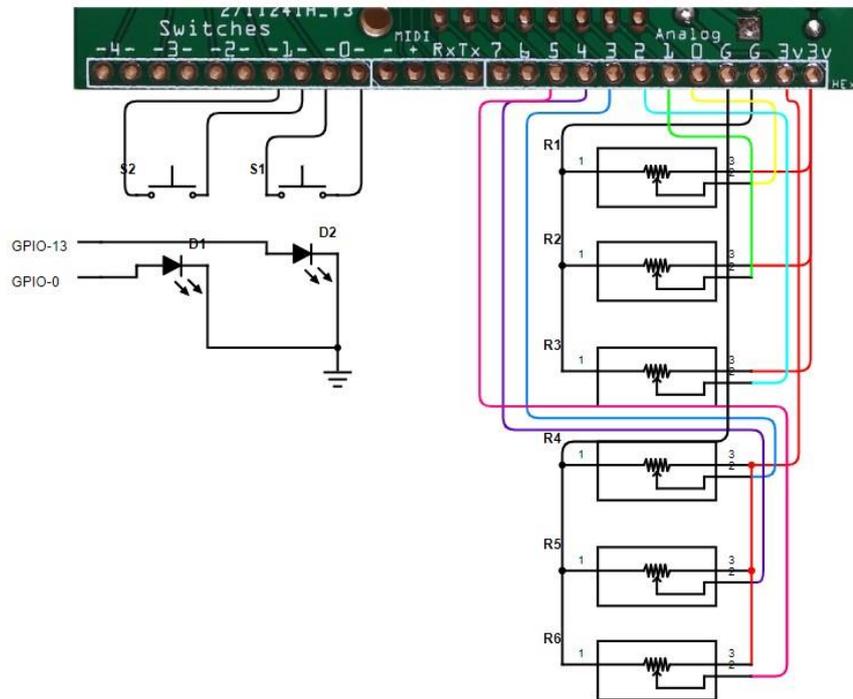


Ilustración 17. Diagrama conexionado controles analógicos.

Hice el conexionado con unos *jumpers* de tal manera que era muy simple conectar los componentes directamente a los pines disponibles en la PCB y modifiqué algunos cables para poder soldar el otro extremo a los controles externos (Ilustración 18).

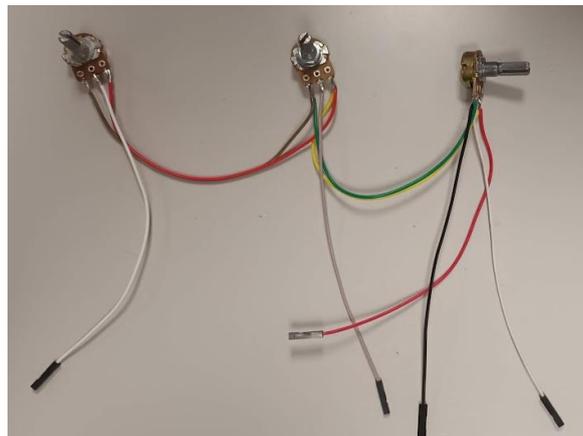


Ilustración 18. Conexionado de potenciómetros en serie.

4.3 Instalación del software

Como bien he comentado anteriormente el proyecto funciona sobre una Raspberry Pi, por lo que siguiendo con las instrucciones disponibles en:

https://www.treefallsound.com/wiki/doku.php?id=software_installation_64-bit

procedo a instalar todo el software necesario en una tarjeta microSD desde mi portátil para poder insertarla luego en la RPi4. Para poder completar el proceso es necesario disponer de lo siguiente:

- Tarjeta microSD con un mínimo de 8GB, recomendable que sea de clase 10.
- Router WiFi conectado a Internet.
- Ordenador con WiFi y lector de tarjetas SD.
- Raspberry Pi Imager instalado.
- Cliente SSH, en mi caso utilicé PuTTY para Windows 10.

Para instalar el sistema operativo en la microSD se utiliza el RPi Imager, se selecciona *Raspberry Pi OS Lite (64-bit)* y se configuran las opciones avanzadas del siguiente modo:

- hostname = pistomp
- Seleccionar "Enable SSH" ,"Use password authentication" y "Set username and password"

De esta manera podremos acceder a la RPI de forma segura y remota desde nuestro ordenador haciendo uso del protocolo SSH [6] configurando el usuario:

- Username = pistomp
- password = *****

Para que la RPI pueda conectarse automáticamente a nuestra red local debemos darle las credenciales de acceso del router.

- Activar la opción "Configure wireless LAN"
- Introducir SSID y contraseña del router

Finalmente se puede guardar y escribir el SO en la tarjeta microSD. Una vez finalizado el proceso se inserta la tarjeta en la RPi y por primera vez se puede alimentar.

Es muy importante no insertar o quitar la tarjeta microSD de la RPi mientras este encendida ya que se podrían corromper los archivos o el SO.

Para poder conectarse a la RPI de forma inalámbrica hay que descubrir la dirección IP asignada a la misma. Existen programas como AngryIP o Ping para ello, yo lo hice accediendo al panel de configuración del router.

Ahora se puede ejecutar el cliente SSH desde el ordenador utilizando esa dirección IP y tendremos que introducir el usuario y contraseña que configuramos anteriormente en RPi Imager (Ilustración 19).

```
login as: pistomp
pistomp@192.168.0.27's password:
Linux piREMIX 5.15.65-rt49-v8+ #2 SMP PREEMPT_RT Sat Mar 25 05:11:57 PDT 2023 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 27 12:12:07 2023 from 192.168.0.19
pistomp@piREMIX:~$
```

Ilustración 19. Conexión a piRMX mediante PuTTY.

Ahora que ya estamos conectados a la RPi podemos empezar con la instalación del software específico de pi-Stomp. Lo haremos a través del repositorio disponible en Github, y para ello se deben ejecutar los siguientes comandos:

```
sudo rpi-update

sudo apt update --fix-missing && sudo apt install -y git

git clone https://github.com/TreeFallSound/pi-stomp.git

cd pi-stomp

nohup ./setup.sh > setup.log | tail -f setup.log
```

Si todo va bien en unos 30 minutos el sistema se reinicia automáticamente y se puede leer en la pantalla “pi-Stomp!”. Después se inicia la pedalera de efectos por defecto y podemos ver los controles principales de la interfaz gráfica en la pantalla LCD (Ilustración 20).

4.3.1 Interfaz gráfica de piRMX en pantalla LCD

La interfaz gráfica que presenta la piRMX se muestra en la Ilustración 20. La información que presenta es la siguiente:

- 1- Pedalera seleccionada
- 2- Preset de la pedalera
- 3- Barra de estado
 - a. WIFI: gris conectado a red local, naranja Pi como punto de acceso
 - b. Bypass canal izquierdo, apagado la señal no es procesada, en verde la señal es procesada.
 - c. Ajustes generales, pulsar para acceder.

- 4- Controles analógicos mapeados.
- 5- Conjunto de pedales, si esta coloreado el efecto está activado.
- 6- Botones mapeados.



Ilustración 20. Pantalla principal piRMX.

4.3.2 Configuración de controles analógicos externos

Dado que los controles analógicos externos son completamente personalizables hay que acceder a un archivo para configurarlos correctamente definiendo el tipo de control que se usa, ya sean botones o potenciómetros, el mensaje MIDI que se activa, y otros parámetros opcionales.

La forma más fácil de acceder al mismo y editarlo es a través de un escritorio remoto, en este caso haciendo uso de VNCViewer [7]. Lo primero es asegurarse de que tanto el ordenador cliente como la RPi están conectadas a la misma red y conectarse a la RPi mediante un servidor SSH como hicimos anteriormente. Una vez conectado podemos descargar el servidor RealVNC en la RPi mediante el siguiente comando:

```
sudo apt install realvnc-vnc-server
```

Ahora debemos activar el acceso a través de VNC entrando en el menú de *“Interface Options”* haciendo uso del siguiente comando:

```
sudo raspi-config
```

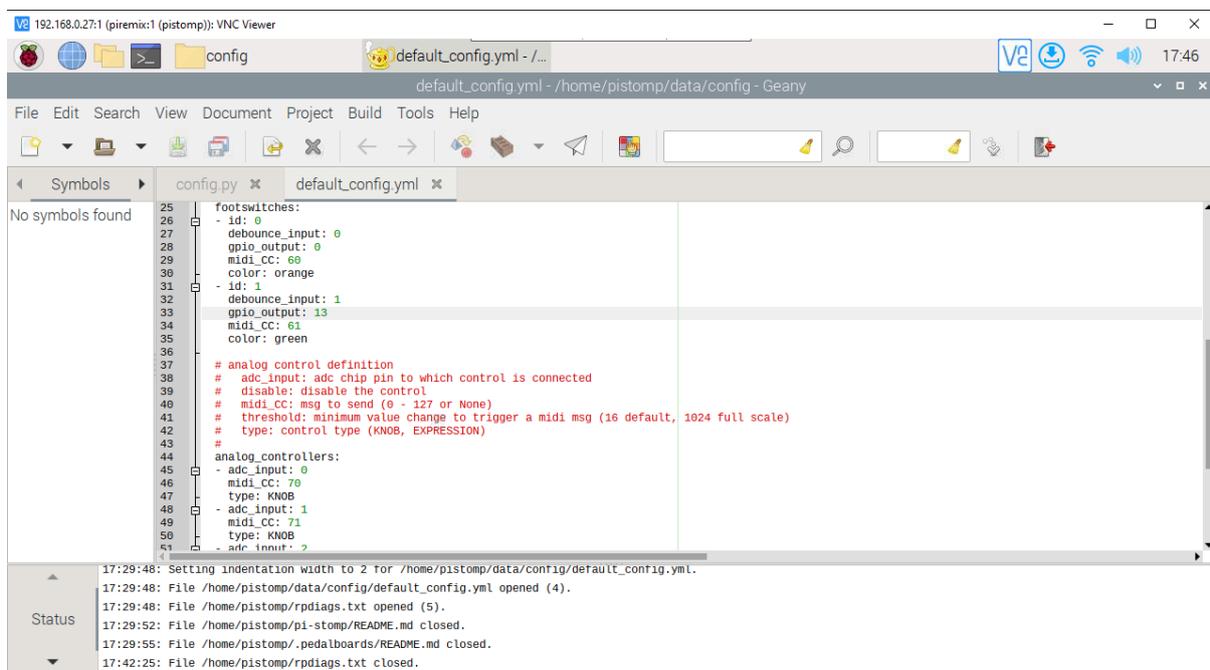
Para iniciar el servidor VNC en la RPi introducimos:

```
$vncserver :1 -geometry 1280x720
```

Ya podemos conectarnos desde la parte del cliente usando la dirección IP de la RPi y las credenciales de acceso del usuario “pistomp” que hemos configurado anteriormente. Deberíamos de poder ver el escritorio de la RPi desde el cliente, y a través del explorador de archivos podemos acceder al archivo de configuración que se encuentra en el siguiente directorio:

```
~/home/pistomp/data/config/default_config.yml
```

El propio archivo (Ilustración 21) contiene información sobre el formato para definir los objetos, en este caso yo tuve que definir dos botones (*footswitches*) y seis potenciómetros (*knobs*). Es muy útil tener a mano el esquema del conexionado de los controles externos para tener claro los pines de conexionado que se están usando para cada control. Una vez terminado guardamos el archivo y reiniciamos la RPi.



The screenshot shows a VNC Viewer window titled "192.168.0.27:1 (piremix:1 (pistomp)): VNC Viewer". The main window displays a file editor with the file "default_config.yml" open. The editor shows a YAML configuration file with the following content:

```
25 footswitches:
26   - id: 0
27     debounce_input: 0
28     gpio_output: 0
29     midi_cc: 60
30     color: orange
31   - id: 1
32     debounce_input: 1
33     gpio_output: 13
34     midi_cc: 61
35     color: green
36
37 # analog control definition
38 #   adc_input: adc chip pin to which control is connected
39 #   disable: disable the control
40 #   midi_cc: msg to send (0 - 127 or None)
41 #   threshold: minimum value change to trigger a midi msg (16 default, 1024 full scale)
42 #   type: control type (KNOB, EXPRESSION)
43 #
44 analog controllers:
45   - adc_input: 0
46     midi_cc: 70
47     type: KNOB
48   - adc_input: 1
49     midi_cc: 71
50     type: KNOB
51   - adc_input: 2
```

The status bar at the bottom shows the following messages:

```
17:29:48: Setting indentation width to 2 for /home/pistomp/data/config/default_config.yml.
17:29:48: File /home/pistomp/data/config/default_config.yml opened (4).
17:29:48: File /home/pistomp/rpdiaqs.txt opened (5).
17:29:52: File /home/pistomp/pi-stomp/README.md closed.
17:29:55: File /home/pistomp/pedalboards/README.md closed.
17:42:25: File /home/pistomp/rpdiaqs.txt closed.
```

Ilustración 21. Archivo de configuración de controles analógicos.

5. PRUEBAS FUNCIONALES

Como en cualquier prototipo, hay numerosas cosas que pueden salir mal o no funcionar correctamente, por eso decidí hacer un par de pruebas para comprobar poco a poco las funcionalidades y calidad de sonido de piRMX.

5.1 Prueba en el laboratorio de instrumentación

Con el hardware y software funcionales me dispuse a realizar algunas pruebas en el laboratorio de instrumentación de la EPSG (Ilustración 22). Lo primero que quise comprobar era la salida de audio estéreo generando sonidos desde el propio piRMX, es decir sin conectar nada en la entrada, y analizar la salida a la vez que probaba la respuesta de los controles analógicos. Mi idea era generar un tono puro desde la interfaz gráfica y conectar los auriculares a la salida, como se les pueden cambiar los transductores tienen la posibilidad de desconectar cada uno de los canales del auricular, de esta manera podía escuchar un canal por uno de los transductores, y el otro canal lo conecté a un osciloscopio para poder comprobar el comportamiento y la calidad de la señal.

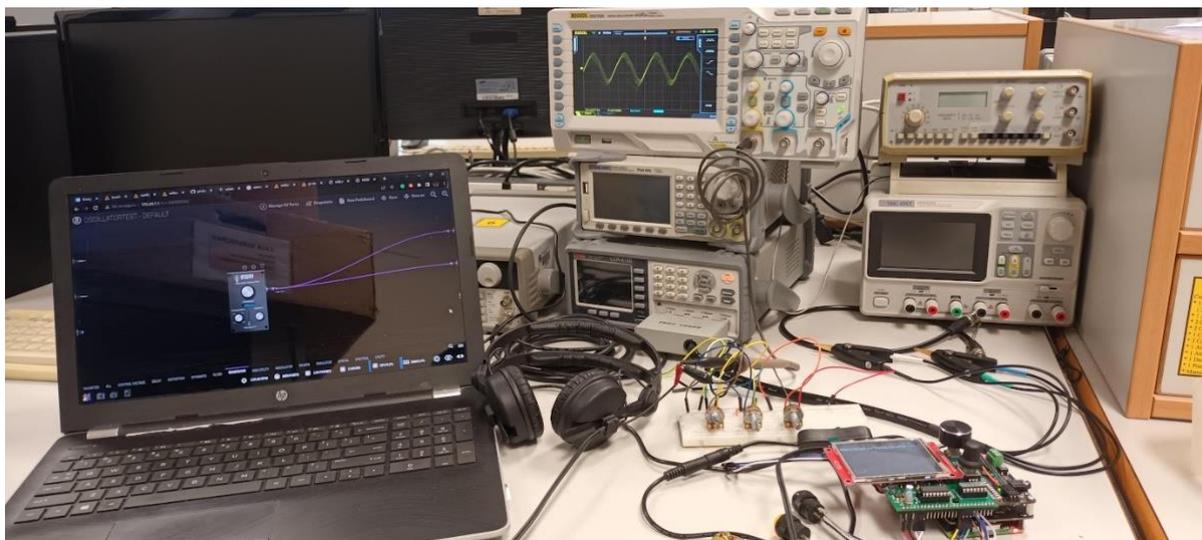


Ilustración 22. Prueba en laboratorio de instrumentación de la EPSG.

5.1.1 Asignación de controles analógicos en MOD-GUI

Con el conexionado hecho, lo primero que había que hacer era configurar el pedal que generaba los diferentes sonidos y mapear sus controles analógicos en la interfaz gráfica web de MOD, para ello debemos conectarnos al piRMX desde el ordenador introduciendo la dirección IP en un navegador web (preferiblemente Chrome). Se puede ver la dirección IP de piRMX navegando con el encoder hasta el icono de ajustes y presionando el encoder, después accedemos al menú de “System info”, donde podremos ver la dirección IP y en caso de no disponer de un router podemos habilitar el *Hotspot* de la RPI para conectarnos directamente

al módulo de efectos. Introducimos la IP de piRMX en el navegador y ya podemos ver la GUI de MOD en el ordenador.

Una vez conectado se crea una nueva pedalera en blanco, y desde el menú deslizable de la parte de abajo se puede buscar y arrastrar los diferentes pedales, ya sean efectos, generadores de sonido, secuenciadores, filtros, etc. Entonces se selecciona un generador de ondas simple y se conecta la salida del generador a ambos canales de salida. En la esquina superior derecha de cada módulo hay un pequeño símbolo de un engranaje para poder configurarlo, y en este caso mapear los controles MIDI que hemos configurado anteriormente en el archivo **default_config.yml**.

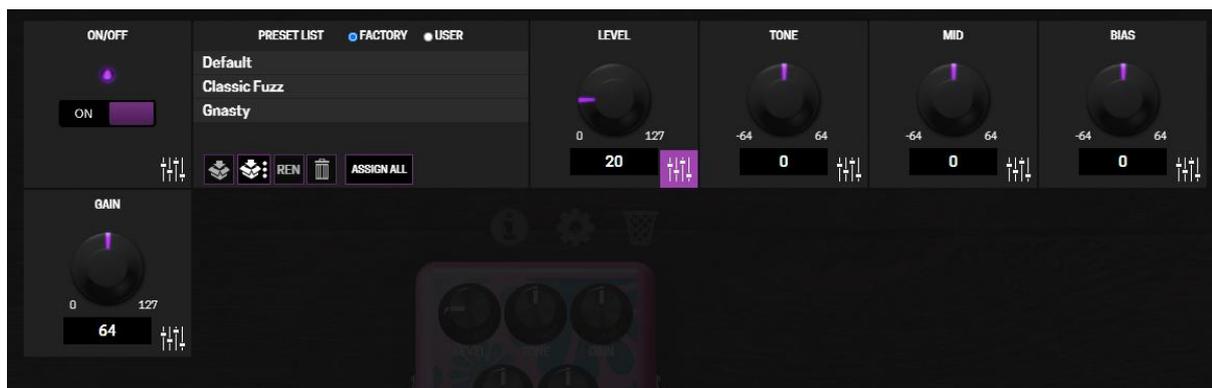


Ilustración 23. Asignación de parámetros a control MIDI.

El mapeado es muy sencillo, dentro de los ajustes del módulo sólo hay que seleccionar el parámetro del efecto que queremos asignar mediante MIDI, ya sea el botón de encendido/apagado, la frecuencia del sonido, la resonancia, y acto seguido mover o presionar el control analógico con el que queremos controlar ese parámetro (Ilustración 23).

Con el módulo que genera ondas senoidales configurado y con los controles analógicos asignados pude comprobar que los controles funcionaban correctamente como esperaba y en el osciloscopio pude comprobar que la onda generada se correspondía con la onda pura senoidal generada en la interfaz web MOD-GUI, no apreciándose ni ruido ni distorsión. Cuando apagaba el módulo que generaba el sonido no había ningún ruido de fondo así que estaba contento con el resultado de la prueba, todo funcionaba como esperaba.

5.2 Prueba en un entorno real con una mesa de mezclas para DJ

El siguiente paso era comprobar el funcionamiento del equipo en un sistema real, haciendo uso de una mesa de mezclas de DJ, conectando tanto la entrada como la salida de audio del módulo de efectos y configurando los controles analógicos para controlar dos efectos de sonido.

El sistema que monte para hacer esta prueba fue con dos tocadiscos y una mesa de mezclas que tenía conexiones de envío y retorno para efectos. En la Ilustración 24 se muestra un diagrama del conexionado.

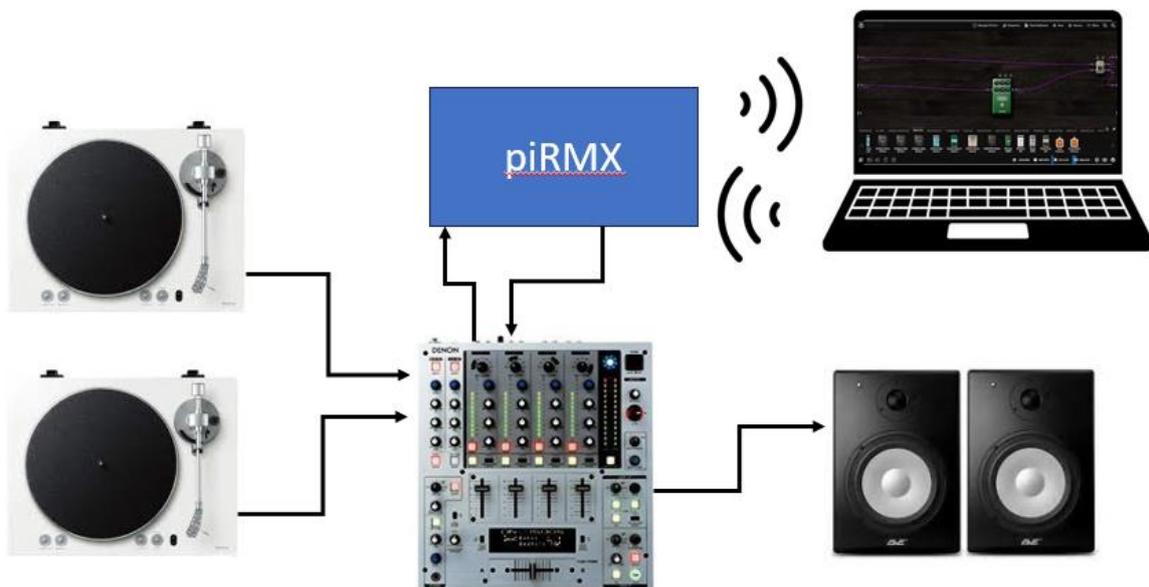


Ilustración 24. Esquema conexionado de piRMX en entorno real.

El ordenador solo es necesario para poder configurar los diferentes efectos y pedales, modificar los parámetros por defecto y asignar los controles analógicos, se guardan los cambios y ajustes, y una vez hecho esto no es necesario conectarse a piRMX ya que funciona de forma independiente.

Para esta prueba cree una nueva pedalera o escena con dos efectos, un *reverb* y un *phaser*, asigné a cada uno un botón para activar o desactivar el efecto y tres parámetros a los potenciómetros. El conexionado de los pedales en MOD-GUI lo hice en cascada, es decir, desde la entrada analógica de audio hacia el *reverb*, después al *phaser*, y de ahí a la salida analógica. Todo esto en estéreo.

In >> Reverb >> Phaser >> Out

Antes de tan siquiera reproducir la música desde los tocadiscos se podía apreciar un ruido de fondo muy molesto y a un nivel considerablemente alto; parecía un ruido digital con un pequeño “click” cada segundo. Asumiendo el problema del ruido me dispuse a seguir con la prueba y el comportamiento del módulo era el esperado, los controles analógicos respondían perfectamente sin ningún retardo entre la activación o desactivación de los efectos y estos se aplicaban correctamente aun que para esta prueba tenía que ajustar manualmente el BPM desde la interfaz gráfica del ordenador en función de la música que estaba reproduciendo desde los tocadiscos. También pude comprobar una de mis mayores preocupaciones, la latencia del piRMX en un entorno real en el cual se mezclan señales que son procesadas a través del módulo de efectos y otras que no son procesadas, no se apreciaba ninguna latencia entre ellas así que esto fue un gran alivio para seguir con el desarrollo del módulo.

A la hora de mezclar canciones, una de las canciones está siendo procesada por piRMX y sale por la salida master de la mesa hacia los altavoces, mientras tanto escucho la otra canción que no está siendo procesada en los auriculares y poco a poco voy tocando y ajustando el tempo hasta que ambas están sincronizadas y así poco a poco puedo introducir la canción que está entrando. Durante todo este proceso es crítico que no haya ningún retardo entre las señales ya que, de no ser así, aunque a través de los auriculares parezca que están sincronizadas, al introducir la nueva canción y subir el volumen se apreciaría un descuadre entre las dos canciones que resultaría en un sonido desacompañado y desagradable. Tras realizar un par de mezclas estaba más que satisfecho con la no apreciación de latencia.

Era muy entretenido y divertido poder tener tanto control sobre los parámetros de los efectos aplicados a las señales de audio, pero seguía teniendo un gran problema de ruido molesto cuando las señales de audio pasaban por el piRMX.

5.2.1 Problema de ruido al procesar señales de entrada

Para poder solucionar el problema del ruido el primer paso era identificar de dónde provenía. En la anterior prueba no había ningún ruido así que supuse que el origen del ruido estaba entre la entrada de hardware TRS y el software de procesamiento del audio. Empecé a hacer algunas pruebas en el software, cree una nueva pedalera y me di cuenta de que si desconectaba virtualmente la entrada de audio de la salida el ruido desaparecía por lo que la anterior suposición era cierta, el ruido estaba en la entrada de audio. Dicho esto, me dispuse a analizar un poco más a fondo el ruido con algunos de los plugins y pedales de MOD-GUI. Conecté un vúmetro y un analizador de espectro a cada uno de los canales y pude ver que el ruido entraba a un nivel de unos -40 dBFS y tenía un pico por debajo de los 20 Hz. El ruido era idéntico en ambos canales.



Ilustración 25. Análisis de ruido en MOD-GUI.

Lo primero que se me vino a la cabeza fue que podía venir de la soldadura que hice en los *jumpers* entre la PCB de pi-Stomp y la tarjeta de sonido que había tenido que cortar y alargar en el montaje del hardware (4.2.2 Montaje del conjunto de placas PCB), así que me dispuse a

cambiarlos por unos jumpers lo suficientemente largos para hacer la conexión. Era un poco complicado ya que uno de los extremos de estos jumpers era de difícil acceso porque estaba entre la PCB y la pantalla LCD y apenas hacía medio centímetro entre ellos, utilicé una pinza de precisión para poder desconectar este extremo de los *jumpers* y poder conectar los nuevos cables. Hecho esto volví a conectar todo y hacer la prueba de sonido, pero el ruido seguía siendo el mismo.

La siguiente posible causa que tuve en cuenta era la fuente de alimentación. Investigando en Internet vi que en este tipo de proyectos donde se combina audio con electrónica es bastante probable que el ruido se deba a que algunos de los armónicos de la red eléctrica se cuelen a través de la fuente de alimentación [8]. Me parecía raro porque en el espectro del ruido no se veían picos en los múltiplos de 50 Hz pero bueno decidí probar a cambiar la fuente para descartar este problema. Utilicé una fuente de alimentación programable existente en el laboratorio de instrumentación de la EPSG, modelo *SPD3303x-E* de la marca *SIGLENT*, pero el ruido no desapareció, así que efectivamente pude descartar que el ruido viniese de la fuente que estaba utilizando desde el principio.

Otra idea que me vino a la cabeza fue que el amplificador operacional estuviese dañado ya que estos componentes son muy sensibles y como tenía otro de repuesto lo cambié, pero tampoco cambió nada en el comportamiento del ruido.

En este punto ya estaba un poco perdido y no sabía por dónde seguir investigando así que me dispuse a analizar el ruido de otra forma para poder tener un análisis con mejor resolución que antes. Conecté a la salida de audio del piRMX a mi ordenador para poder tomar una muestra de audio con el software Adobe Audition, y obtuve el análisis del espectro mostrado en la Ilustración 26.

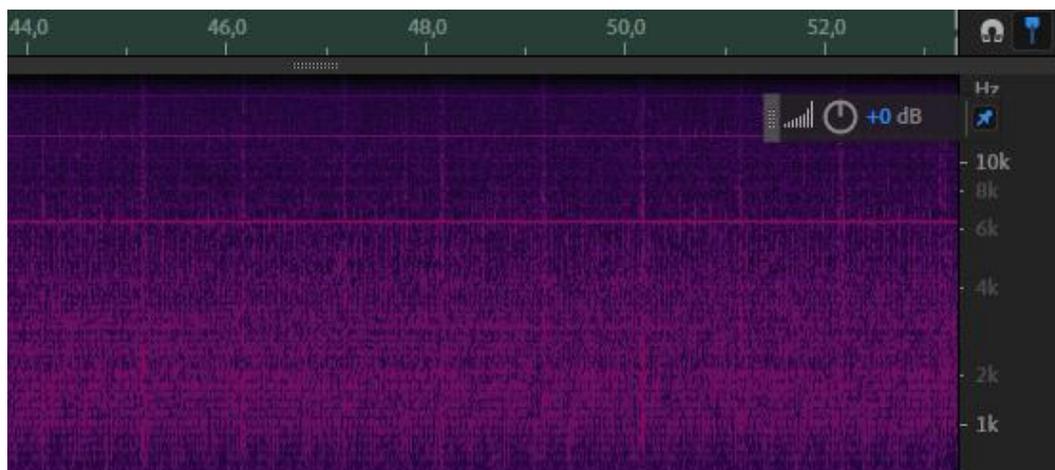


Ilustración 26. Espectrograma del ruido en Adobe Audition.

De este análisis saqué como conclusiones que el ruido era bastante denso por debajo de los 4 kHz y había picos en 6,3 kHz y en 12,6 kHz, había un pequeño “click” cada segundo. Parecía de origen digital pero tampoco me dio muchas más pistas de su procedencia, así que me puse en contacto con el creador del proyecto de TreeFallSound, Randall, y le envié un email detallando todos los pasos anteriores con las muestras de audio y las imágenes. Me confirmó

que debía de ser de procedencia digital y que seguramente se colaba por la alimentación a través de la RPi [9]. Me fue guiando para comprobar que los componentes de la PCB de pi-Stomp estaban bien conectados y orientados y hacer un seguimiento de la señal de audio a través de esta pero tampoco encontré el origen del problema. En una de sus respuestas me dijo que probase a quitar el condensador C5 ya que en algún otro caso ya le había solucionado el problema y efectivamente así fue, el ruido desapareció instantáneamente. Era irónico porque me dijo que en su diseño original este condensador tenía como función reducir el ruido en el regulador de tensión de 5 voltios de la RPi.

Volví a montar el módulo de efectos con el sistema de sonido descrito anteriormente y por fin el dolor de cabeza del ruido desapareció y el piRMX se podía utilizar con la mesa de mezclas sin introducir ruido al aplicar efectos en las señales procesadas. Estuve unas dos horas mezclando y jugando con diferentes efectos y configuraciones del envío y retorno de estos y por fin pude disfrutar de un módulo funcional y profesional a la hora de pinchar, era un poco incómodo por el momento ya que los controles analógicos estaban al descubierto y había que utilizar las dos manos para manipularlos, así que el siguiente paso era introducir todo el hardware en una caja metálica que fuese funcional.



Ilustración 27. Uso de piRMX con otros equipos de audio usados por DJs.

6. MONTAJE DENTRO DE UNA CAJA METÁLICA

6.1 Primer boceto del diseño

Para diseñar la caja ya tenía en mente la idea principal basándome en el diseño original de TreeFallsound (Ilustración 4), los controles analógicos estarían en la parte de abajo mientras que la pantalla, el encoder y el control del volumen de salida estarían arriba. Como bien se vio en la sección de diseño del hardware (sección 4.1) tenemos dos efectos, cada uno con tres potenciómetros y un botón. Hice varios bocetos sobre papel con diferentes configuraciones de los controles analógicos teniendo en cuenta la utilidad de cada uno de ellos y usabilidad a la hora de mezclar hasta que decidí quedarme con el diseño de la Ilustración 28.

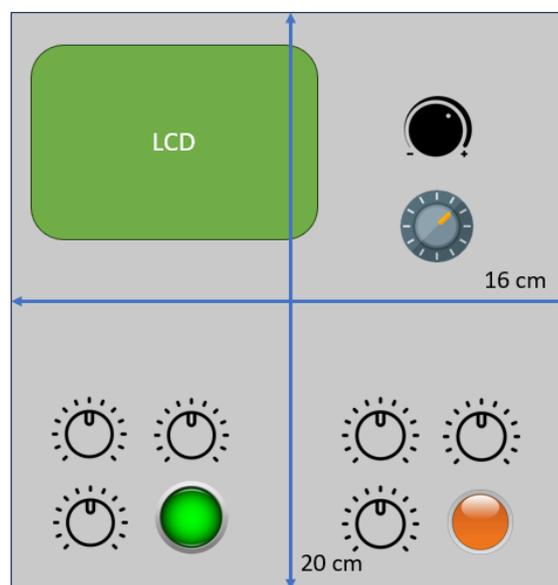


Ilustración 28. Primer diseño del layout piRMX.

El boceto estaba aproximado a la escala real de los componentes, así que tome las medidas del alto y ancho para después poder buscar una caja adecuada al diseño. Hay que decir que añadí unos 2-3 centímetros a cada medida para tener un poco más de espacio y quizás en un futuro poder añadir más controles analógicos.

-Alto (eje Y de la ilustración 27): 20 cm

-Ancho (eje X de la ilustración 27): 16cm

Para la profundidad tuve en cuenta que la pantalla, el volumen y el encoder tenían que estar a ras de superficie y desde ahí pude medir hasta la placa que quedaba más profunda, de manera que el conjunto de placas tenía una profundidad de 4,5 centímetros. Más tarde pensé que a la hora de usar el piRMX en una cabina de DJ lo más conveniente era que tuviese la misma altura que el resto de los equipos de audio, así que me puse a investigar un poco por Internet, y tras mirar varios manuales me di cuenta de que todos los equipos tienen una

profundidad de unos 7-8 centímetros, así que esta sería mi referencia a la hora de buscar una caja para el hardware.

En cuanto al material de la caja estaba claro que tenía que ser metálica para evitar las interferencias electromagnéticas que podrían manifestarse como ruido, distorsión o zumbidos en las señales de audio.

Tras varias horas investigando por Internet en foros y páginas web de proyectos caseros de audio [10] me decidí por las cajas de la empresa *Hammond Manufacturing*. Están hechas de una aleación de aluminio mediante un proceso de fundición a presión y tienen un precio muy razonable teniendo en cuenta la gran calidad del producto, además de ofrecer una gran variedad de tamaños y acabados. Finalmente encontré una caja con un acabado negro anodizado, muy elegante, que se acercaba a las medidas que buscaba, y eran las siguientes:

-Alto: 23 cm

-Ancho: 17 cm

-Profundo: 7 cm

En cuanto al espesor de la caja es de entre 2-3 mm.

Enlace a la caja elegida: <https://www.hammmfg.com/part/1457U2201BK?referer=1204> .

6.2 Problema de incompatibilidad en los conectores del panel trasero

Con la caja en mis manos el reto era conseguir plasmar las medidas exactas de la pantalla, botones y conectores en esta. Hablando de conectores me volvió a la cabeza el problema que comenté en el montaje del hardware (4.2.2 Montaje del conjunto de placas PCB) por la incompatibilidad de las profundidades de los conectores de la RPi los de la PCB de TreeFallSound (Ilustración 29).



Ilustración 29. Conectores traseros piRMX

Tras darle muchas vueltas se me ocurrieron varias opciones.

OPCIÓN 1

Dejar los conectores como estaban y hacer un agujero algo más grande en la parte de atrás de la caja, de esta forma los conectores de la RPi estarían a ras de la caja y la alimentación y los conectores *jack* TRS se quedarían escondidos dentro de la caja con un agujero para tener acceso a ellos. El acabado sería poco profesional y además por esos agujeros entraría un montón de polvo y partículas dentro de la caja. Y, por si fuera poco, al no tener los conectores TRS atornillados a través de la caja la resistencia mecánica del hardware dentro de la caja se vería comprometida cada vez que conectásemos los cables de audio ya que sólo se uniría a esta a través del encoder y el potenciómetro de volumen.

OPCIÓN 2

Había visto por internet unos cables *jumper*s para el conector de 40 pines de la RPi, de esta manera podría desatar la RPi del conjunto e instalar todos los conectores en el mismo plano. Me gustó la idea porque así podría dejar la RPi en cualquier posición de la caja, por ejemplo, en una esquina, sin que molestase a la hora de montar el resto de los componentes. Era buena idea hasta que me di cuenta de que el módulo LCD también era más profundo que los conectores así que esta opción quedó completamente descartada ya que tendría que desoldar la pantalla, buscar la manera de reubicarla con unos cables y pegarla a la caja. Eran muchas modificaciones complejas para solucionar el problema.

OPCIÓN 3

Tratar de desoldar los conectores *jack* TRS de la PCB y así poder utilizar las tuercas que llevan para atornillarlos a través de unos agujeros a la parte de atrás de la caja, de esta manera las placas no sufrirían al conectar y desconectar los cables de audio. El único problema es primero desoldarlos ya que los pines de los conectores TRS tienen difícil acceso con el LCD, teniendo cuidado de no dañar ningún componente ni cable al hacerlo. Y después solo habría que conectar unos cables para hacer la conexión desde los conectores que estarían ajustados al borde de la caja hasta los puntos de conexión de la PCB. Esta es la mejor opción desde el punto de vista de la resistencia mecánica del conjunto hardware-caja y la que menos tiempo requeriría.

Manos a la obra, lo primero era desmontar el conjunto de todas las placas y quitar los conectores para poder trabajar más cómodo sobre la placa de pi-Stomp. Como no tenía acceso a los pines soldados de los conectores TRS tuve que desoldar los 9 pines del módulo LCD para poder quitarlo, fue una tarea bastante más compleja de lo que pensaba. Tuve que aplicar una gran cantidad de estaño uniendo todos los pines de tal forma que pudiese calentar

y fundir los 9 pines a la vez. Ya tenía acceso a todos los pines de la PCB de pi-Stomp y pude desoldar los dos conectores TRS y ya que era fácil también quité el conector de la alimentación, así podría ponerlos todos a ras del panel trasero de la caja. Con los conectores libres, corté un par de cables con una longitud de 10cm y los soldé en sus correspondientes pines para poder moverlos donde quisiese. Resulto ser mucho más trabajo del que esperaba, pero conseguí volver a conectar, soldar y montar todos los componentes. Comprobé las conexiones con el polímetro y lo puse en funcionamiento para comprobar las conexiones como se puede observar en la Ilustración 30.

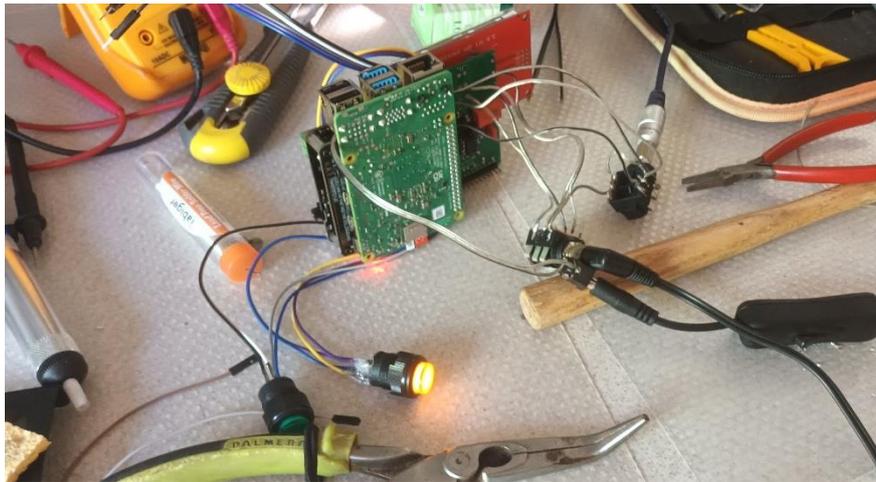


Ilustración 30. Funcionamiento tras alargar conectores traseros.

6.3 Maquetación y montaje en la caja

Para proseguir con el montaje, me ayudé de un calibre para poder tomar todas las medidas necesarias para hacer los correspondientes agujeros en la caja. De esta manera completé el diseño anterior de la Ilustración 28 con mucho más detalle, como se puede ver en la Ilustración 31. Después procedí a plasmar las medidas en la caja metálica con la ayuda de varios instrumentos de medida además del calibre como reglas y escuadras para procurar ser lo más preciso posible, calculando muy bien los centros de todos los agujeros necesarios.

Los cortes redondos eran fáciles ya que sólo necesitaba la broca del calibre necesario, pero para los cortes rectangulares tuve que apañármelas haciendo agujeros los más cercanos posibles y después lijar a mano el aluminio para conseguir los rectángulos deseados. Después con una lija más fina terminé de afinar los bordes y tenía todos los recortes finalizados.

Primero instalé los botones y los potenciómetros en cada uno de sus agujeros asegurándome que estaban bien apretados y no se moverían con el uso. Quería conectar los controles analógicos con las placas PCB fuera para que fuese más cómodo pero los cables no eran lo suficientemente largos, así que primero instalé las placas, con el control de volumen, el

encoder y la pantalla, y después hice el conexionado en los pines correspondientes de la PCB de pi-Stomp con mucho cuidado y la ayuda de unas pequeñas pinzas.

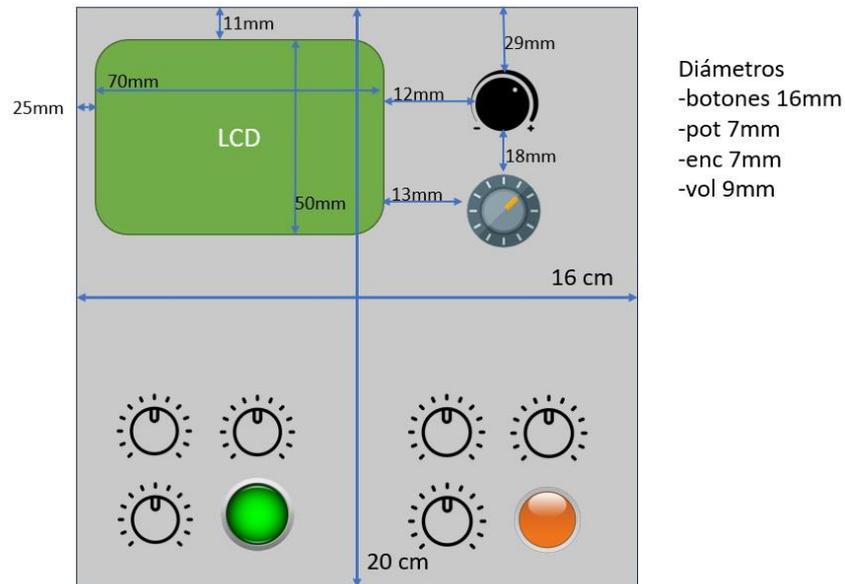


Ilustración 31. Diseño con medidas del panel frontal.

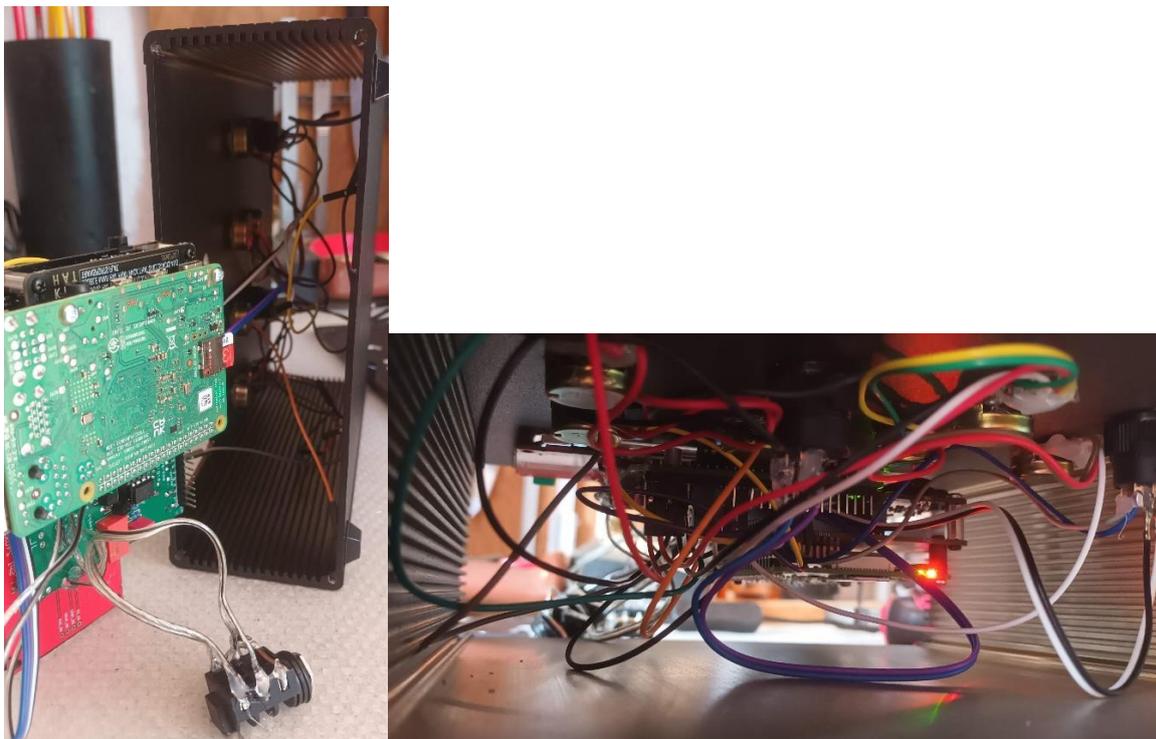


Ilustración 32. Montaje de hardware en la caja metálica.

Con todo el hardware montado en la caja sólo quedaba hacer una ultima prueba de funcionamiento y cerrar la caja por ambos extremos. ¡El piRMX estaba listo!

7. CONCLUSIONES Y POSIBLES MEJORAS

En este proyecto se ha cumplido el objetivo de desarrollar e implementar el módulo de efectos de sonido basado en el proyecto de Tree Fall Sound adaptándolo a las necesidades de un DJ. A partir de la documentación disponible e investigando en Internet he sido capaz de hacer realidad el módulo de efectos personal y profesional que voy a poder utilizar juntamente con mis equipos de audio para mezclar en directo, teniendo un mayor control sobre cada uno de los efectos y con una personalización al detalle gracias al software de MOD Audio y a los controles analógicos que he decidido utilizar.

He ampliado mis conocimientos en el ámbito de la electrónica aplicada al audio y he podido comprobar las dificultades que conlleva el diseño conjunto de hardware y software a pesar de que el proyecto inicial ya estaba muy avanzado y preparado para su reproducción en cualquier parte del mundo. Durante el desarrollo me he enfrentado a diversas dificultades y problemas que he tenido que analizar, investigar, solucionar y probar combinando mis conocimientos con la ayuda de otras personas y toda la información disponible en Internet.

He abordado el proyecto de forma personal, lo cual me ha hecho invertir un poco de dinero y una gran cantidad de horas en él de forma muy fácil y sin que me costase ponerme manos a la obra. Me podría haber organizado algo mejor y haber establecido unos objetivos o unas metas intermedias que seguramente me hubiesen ayudado en los momentos que el trascurso del proyecto se bloqueaba o veía condicionado por mi u otros factores externos como envíos de material.

Todavía no he podido exprimir al máximo la capacidad de la CPU de la RPi, pero de momento no he conseguido utilizar más de un 50 % de la misma utilizando solo cinco o seis pedales, lo cual me hace pensar que igual se hubiese podido simplificar y reducir el precio del proyecto con un modelo más básico como por ejemplo la RPi 3A o 3B.

7.1 Trabajos futuros

7.1.1 Detección automática de BPM

Debido a mi inexperiencia en la materia de desarrollo de software, las dificultades encontradas y la falta de tiempo no me han permitido investigar e implementar la funcionalidad de detección de BPM automatizada dentro del módulo de efectos para conseguir unos efectos sincronizados con la música en tiempo real. Lo he tenido en cuenta durante todas las pruebas y tenía la idea de crear un servicio en Python que analizase en tiempo real el BPM y mandase el valor del BPM a través de mensajes MIDI cada 5 segundos al software MOD-Audio. En un futuro me gustaría poder implementar esta funcionalidad, así como otras muchas mejoras.

7.1.2 Personalización de la interfaz gráfica y pantalla táctil

Sería interesante modificar la interfaz gráfica que se visualiza en la pantalla LCD de piRMX, lo primero sería poder visualizar en un pequeño recuadro los BPM a los que está trabajando el módulo de efectos. También echo en falta tener una referencia visual de los valores de cada uno de los parámetros, pero es cierto que en una pantalla tan pequeña es complicado de integrar, quizás para este requerimiento haría falta una pantalla algo más grande.

En el apartado del montaje del hardware (4.2.1 Soldadura de componentes electrónicos en PCB de pi-Stomp) deje los cables de la interfaz SPI del panel táctil sin conectar pero listos para poder ser conectados en un futuro. Sería muy interesante en un futuro poder aprender más sobre esta interfaz para ser capaz de implementarlo dentro del software y ser capaz de navegar a través de los menús de forma más intuitiva. También sería interesante tener la posibilidad de controlar la activación y dos parámetros de un mismo efecto a través de la pantalla táctil.

7.1.3 Añadir más controles analógicos

Durante el desarrollo del proyecto siempre he tenido el pensamiento de que el piRMX tiene mucha más capacidad de procesamiento y tiene todavía algunos conectores para controles analógicos libres, por lo cual es inevitable pensar en añadir más controles analógicos para poder controlar más efectos o ajustes del software. Tras usarlo con otros equipos de DJ, pensé que sería divertido por ejemplo añadir dos botones y un potenciómetro para poder controlar un pedal de MOD que actúa como un *Looper* y así poder aplicar bucles en directo, lo cual añade una infinidad de posibilidades creativas para el DJ.

8. BIBLIOGRAFÍA

[1]

“History of DJing.” [Online]. Available:

<https://www.skilzdjacademy.com/post/2019/02/01/dj-history#:~:text=History%20of%20DJing&text=In%201943%2C%20radio%20DJ%20Jimmy,Ancient%20Shepherds%20in%20Otley%2C%20England.>

[2]

M. Callander, “Thinking About Syncing: Examining the impact of 21 st century DJ technology on the production and performance of Electronic Dance Music.” 31-Mar-2023.

[3]

R. Reichenbach, “DIY Multi-effects by TreeFallSound.” [Online]. Available:

<https://www.treefallsound.com/>

[4]

“MOD Web GUI User Guide - MOD Wiki.” [Online]. Available:

[https://wiki.mod.audio/wiki/MOD_Web_GUI_User_Guide.](https://wiki.mod.audio/wiki/MOD_Web_GUI_User_Guide)

[5]

“Digital Audio Latency Explained.” [Online]. Available:

<https://www.presonus.com/learn/technical-articles/Digital-Audio-Latency-Explained>

[6]

“What is SSH (Secure Shell) and How Does it Work? Definition from TechTarget.” [Online].

Available: [https://www.techtarget.com/searchsecurity/definition/Secure-Shell.](https://www.techtarget.com/searchsecurity/definition/Secure-Shell)

[7]

“Descargue VNC Viewer | VNC Connect.” [Online]. Available:

[https://www.realvnc.com/es/connect/download/viewer/.](https://www.realvnc.com/es/connect/download/viewer/)

[8]

“Raspberry Pi power supply choice.” [Online]. Available:

<https://www.diyaudio.com/community/threads/raspberry-pi-power-supply-choice.326574/>.

[9]

“Raspberry Pi Power Issues, Audio Server project with VolumIO” [Online]. Available:

<http://www.djerickson.com/rpi-power/>.

[10]

“DIYstompboxes.com - Index.” [Online]. Available:

<https://www.diystompboxes.com/smfforum/index.php>.

9. ANEXOS

Anexo I. Relación del trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030.

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS):

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.			X	
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.			X	
ODS 8. Trabajo decente y crecimiento económico.			X	
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.		X		
ODS 12. Producción y consumo responsables.	X			
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.	X			

Descripción de la alineación del TFG con los ODS con un grado de relación más alto:

Objetivo 3: Salud y bienestar. La música tiene un impacto significativo en el bienestar emocional, social y físico de las personas.

Objetivo 9: Industria, innovación e infraestructura. Al desarrollar el módulo de efectos se está contribuyendo a la innovación en la industria musical y a la mejora de la infraestructura utilizada en conciertos en directo.

Objetivo 12: Producción y consumo responsables. El módulo ha sido diseñado y concebido de tal forma que sea reproducible en cualquier parte del mundo por cualquier persona y tiene una vida útil muy larga. Al ser un proyecto abierto, si hubiese un fallo o se rompiese un componente se puede reparar fácilmente esa parte sin necesidad de comprar un módulo de efectos nuevos.

Objetivo 17: Alianzas para lograr objetivos. A lo largo del proyecto se ha colaborado e interactuado con algunos expertos en electrónica aplicada al audio, fomentando la colaboración y el intercambio de conocimientos en busca de soluciones sostenibles.

Anexo II. Diccionario de acrónimos

- ADC: Analog to Digital Converter (Convertidor Analógico a Digital).
- BPM: Beats Per Minute (Pulsaciones Por Minuto).
- CPU: Central Processing Unit (Unidad de Procesamiento Central).
- DAW: Digital Audio Workstation (Estación de Trabajo de Audio Digital).
- DJ: Disk Jockey (Pinchadiscos).
- GUI: Graphical User Interface (Interfaz Gráfica de Usuario).
- GPIO: General Purpose Input/Output (Entrada/Salida de Propósito General).
- LV2: Estándar abierto para plugins de audio.
- MIDI: Musical Instrument Digital Interface (Interfaz Digital de Instrumentos Musicales).
- PCB: Printed Circuit Board (Placa de Circuito Impreso).
- RPi: Raspberry Pi (Una serie de computadoras de placa única).
- SO: Sistema Operativo.
- SPI: Serial Peripheral Interface (Interfaz Periférica en Serie).
- SSH: Secure Shell (Protocolo para acceso remoto a un servidor por medio de un canal cifrado).

Anexo III. Coste

El coste del proyecto se recoge en la siguiente tabla:

Concepto	Importe	Tienda
pcb piStomp y componentes	57,82 €	Randall_pistomp
componentes, cables y alimentación	68,62 €	mouser.es
Raspberry Pi 4B	82,60 €	tiendatec.es
microSD	10,00 €	amazon.es
HiLetgo ILI9341 2.8" SPI TFT LCD pantalla táctil 240 x 320	18,00 €	amazon.es
IQAudio codec zero	27,90 €	tiendatec.es
Hammond aluminium box	66,79 €	digikey.es
	331,73 €	TOTAL

El coste sin tener en cuenta la mano de obra de piRMX es de 331,73 €. Es algo más de lo que tenía pensado cuando empecé con el proyecto, pero teniendo en cuenta el precio de los equipos similares existentes en el mercado es un proyecto asequible para personas con cierto conocimiento técnico y ganas de invertir algo de tiempo en el montaje.

Se podría ajustar el precio haciendo uso de un modelo menos potente de RPi ya que de momento no he sido capaz de utilizar más de un 50% de la capacidad de la CPU, aunque también es cierto que no he usado pedaleras con más de 5 o 6 pedales. Sería interesante hacer un estudio del rendimiento de diferentes modelos de RPi en función del número de pedales que se pueden usar sin comprometer la calidad del audio. También se podrían reducir costes en los componentes y cables buscando otros proveedores más asequibles y haciendo un único pedido a fin de pagar un menor número de envíos.

Anexo IV. Fotos montaje final de piRMX



Ilustración 33. piRMX frontal



Ilustración 34. piRMX trasero