

GUÍA DE CONSTRUCCIÓN

para una máquina de ritmos



OPEN-SOURCE EDUCATIONAL KIT
2024

Escrito por

Andréé Fuentes

Guatemala 2024

Universidad del Valle
de Guatemala

Índice

| | |
|---|----|
| 1. Introducción | 5 |
| 2. Listado de Componentes | 6 |
| 3. Esquemático | 12 |
| 4. Diseño e Impresión del PCB | 14 |
| 5. Diseño e Impresión de la carcasa | 16 |
| 6. Soldadura | 19 |
| 7. Código Fuente | 22 |
| 8. Ensamblaje | 29 |
| 9. Prompts de ejercicios | 32 |
| 10. GitHub | 35 |

1

Introducción

Como amante de la música, la tecnología y la intersección entre ambas, decidí crear este manual con el propósito de que más personas puedan comprender qué conlleva la creación de un instrumento musical electrónico.

Tomando como referencia las icónicas *TR-808* y *TR-909* de la Roland Corporation en años 80, diseñé las distintas partes que componen una máquina de ritmos física, pero programada digitalmente. Este proyecto *Open Source* busca potenciar la creatividad e innovación de cualquier persona que acepte el reto, y evitar la total digitalización de este instrumento pionero de géneros tales como el el EDM y el hip hop.

La OR-1 es una máquina de ritmos con un secuenciador de 16 pasos que permite cargar tus propios samples para programar patrones rítmicos. Cuenta con un filtro pasa altas, un efecto de distorsión y la capacidad manejar varios modos de operación, tales como FreePlay/Programming y RepeatMode. Incluye una interfaz de audio USB y monitoreo por audífonos, además de una pantalla para la visualización de los modos de funcionamiento.

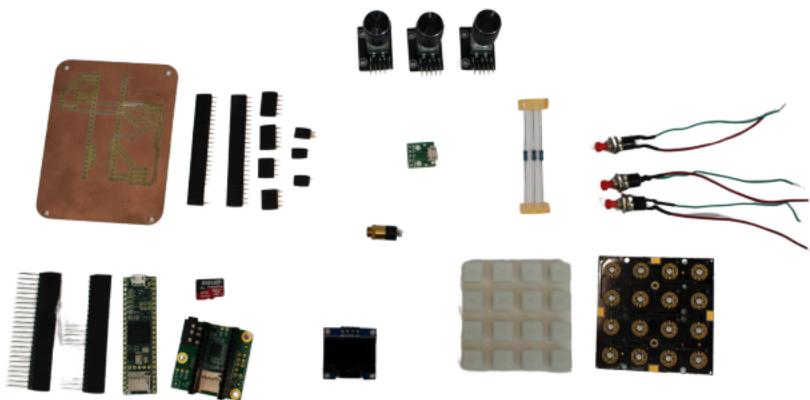
Este manual te guiará paso a paso en el proceso de ensamblaje, soldadura y programación de tu dispositivo, para que puedas empezar a crear música con la OR-1.

Todos los archivos necesarios se encuentran en el GitHub del proyecto, anexado al final del manual.

2

Listado de Componentes

En esta sección, se detallan los componentes necesarios para la construcción de la OR-1 con una breve descripción de cada uno y su función en el proyecto. Esta sirve de referencia para poder asegurarte de tener todo listo antes de comenzar el ensamblaje.



| | | |
|---|--|---|
|  1 x Teensy 4.1 |  1 x Audio Board Rev D |  3 x resistencias 10K Ω |
|  3 x botón de pulsador |  2 x encoder KY-040 |  1 x Adafruit NeoTrellis 4x4 |
|  1 x Adafruit NeoTrellis Silicone Pad |  1 x pantalla i2c 128x64 |  1 x jack 3.5mm stereo |
|  1 x PCB micro USB hembra |  2 x cable micro USB |  2 x header hembra 24 pin corto |
|  2 x header hembra 5 pin corto |  2 x header hembra 4 pin corto |  3 x header hembra 2 pin corto |
|  4 x header hembra 24 pin largo |  1 x kit de 60 cables dupont macho - macho |  1 x tarjeta micro SD |

1. Teensy 4.1

Es el microcontrolador y la parte más esencial de la OR-1. Cuenta con un procesador ARM Cortex-M7 a 600Mhz, 1 MB de RAM y 8 MB de memoria Flash. Está diseñado específicamente para aplicaciones de audio, proporcionando la potencia suficiente para procesamiento de señales digitales complejas. Se programan con el lenguaje del Arduino IDE.

2. Audio Board Rev D

Es la tarjeta de expansión de audio para el Teensy 4.1. Este módulo incluye un códec de audio SGTL5000 que proporciona entrada y salida estéreo de alta calidad. Facilita la adición de capacidades de procesamiento de audio como efectos, filtros y más.

3. Resistencias de $10K\Omega$

Resistencias utilizadas para la configuración de los botones. Estas aseguran que los pines de entrada de los botones se mantengan en un estado bajo cuando los botones no están siendo presionados, previniendo lecturas erróneas debido al ruido eléctrico que puede darse en una placa electrónica.

4. Botones de pulsador

Botones utilizados para las interacciones manuales de la OR-1. Este tipo de dispositivo permiten al usuario enviar señales de entrada al microcontrolador. En la máquina, controlan los modos: Repeat, Freeplay/Programming y Pausa/Play.

5. Encoder KY-040

Los encoders son dispositivos que miden la rotación y precisión angular, es decir, convierten el movimiento del knob en señales digitales que pueden ser interpretadas por el microcontrolador. Sus parámetros pueden ser guardados digitalmente y permiten el control preciso de parámetros como tempo, volumen, filtros y efectos.

6. Adafruit Neotrellis 4x4 con LEDS

Es una matriz de 16 botones que pueden ser iluminados por LEDS RGB. Se comunica con el microcontrolador a través del protocolo I2C. En la OR-1, es el secuenciador de 16 pasos que permite la programación de ritmos.

7. Adafruit NeoTrellis Silicone Pad

Es el pad de silicona para el NeoTrellis 4x4. Proporciona una superficie táctil suave y duradera para los botones del NeoTrellis, facilita y mejora la experiencia de usuario y aporta a la durabilidad del OR-1.

8. Pantalla I2C 128x64

Esta pantalla de visualización permite la visualización de los estados de la máquina en la interfaz de usuario. Se comunica por protocolo I2C con el microntrolador. En ella, se muestra la información crucial como el modo de operación, BPM, y otros parámetros de la OR-1.

9. Jack 3.5mm stereo

Es un conector de salida de audio para audífonos o altavoces. Permite conectar la OR-1 a dispositivos de audio externos, utilizado como monitoreo personal con audífonos con reproducción de sonido en alta calidad.

10. PCB micro USB hembra

Esta placa con conector micro USB funciona como una extensión del puerto micro USB del Teensy 4.1 para poder ser accedido desde afuera de la OR-1 como fuente de alimentación o para su salida de audio USB (compatible con diferentes DAWs).

11. Cable micro USB

Estos cables de conexión para transferencia de datos y energía funcionan para conectar la OR-1 a una computadora o a una fuente de alimentación. Uno de los cables es utilizado dentro de la máquina para puentear entre el micro USB hembra del Teensy 4.1 y la placa hembra en el exterior de la máquina.

12. Header hembra 24 pin corto

Utilizado para soldarse en el PCB de la OR-1 para posteriormente montar el Teensy 4.1.

13. Header hembra 5 pin corto

Utilizado para soldarse en el PCB de la OR-1 para posteriormente montar los encoders.

14. Header hembra 4 pin corto

Utilizado para soldarse en el PCB de la OR-1 para posteriormente montar los dispositivos I2C.

15. Header hembra 2 pin corto

Utilizado para soldarse en el PCB de la OR-1 para posteriormente montar los botones.

16. Header hembra 24 pin largo

Utilizado para soldarse a los pines del Teensy 4.1 y Audio Board Rev D (14 pines). El Teensy 4 con pin largo va montado sobre el pin corto del PCB, y el Audio Board va montado sobre el Teensy.

17. Kit de 60 cables dupont macho - macho

Estos cables son flexibles y fáciles de utilizar. Serán utilizados para los componentes principales, buscando que se puedan montar y desmontar fácilmente de los headers hembra en el PCB de la OR-1.

18. Tarjeta micro SD

Es la memoria externa para el almacenamiento de los samples de la máquina de ritmos. Es introducida en el Audio Board Rev D.

Adicional recomendado: termoencogible para evitar la interconexión entre pines.

*Los headers suelen conseguirse como tiras largas, deben ser cortados para obtener el tamaño deseado.

3

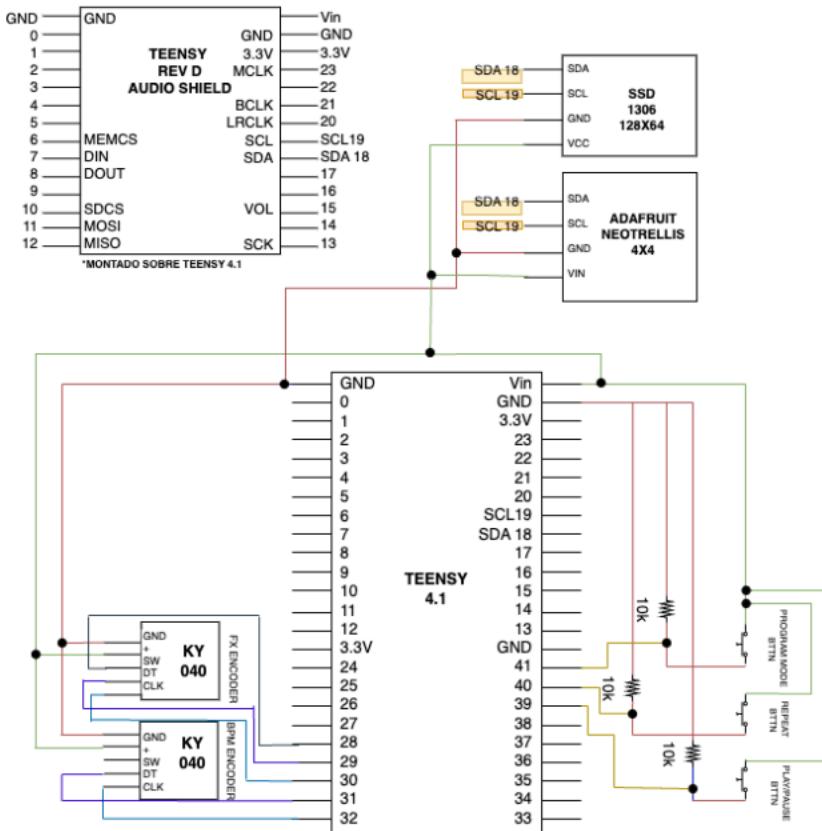
Esquemático

Un esquemático es una representación gráfica que detalla las conexiones entre componentes electrónicos mediante símbolos estándar y líneas representativas. Este facilita la planificación del circuito, permitiendo identificar y corregir errores previos a y durante la construcción.

El esquemático de la OR-1, ya desarrollado, presenta un diseño centrado en el microcontrolador Teensy 4.1, encargado del procesamiento de audio y las operaciones generales. El Audio Board Rev D se monta directamente encima sin necesidad de cableado adicional, y se especifican los pines correctos para el mismo.

La comunicación para la pantalla y NeoTrellis con el Teensy se realiza mediante el protocolo I2C, que utiliza solo dos líneas: SDA para datos y SCL para sincronización. Para simplificar el esquemático y evitar confusión, no se dibujaron las líneas entre estos componentes pero se indica claramente su conexión a los pines SDA 18 y SCL 19 del Teensy. También se detalla la conexión de las resistencias, botones y encoders de manera clara.

Es importante analizar y comprender el esquemático para poder realizar todas las conexiones correctamente para evitar errores que puedan dañar los componentes o afectar el funcionamiento del dispositivo. Para esto, se añade un código de colores de fácil seguimiento.

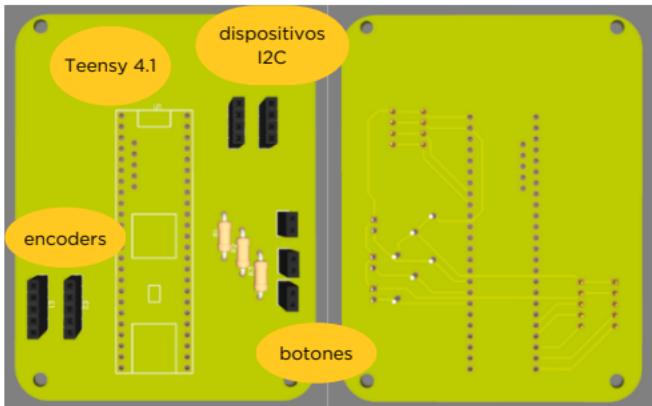


4

Diseño e impresión de la PCB

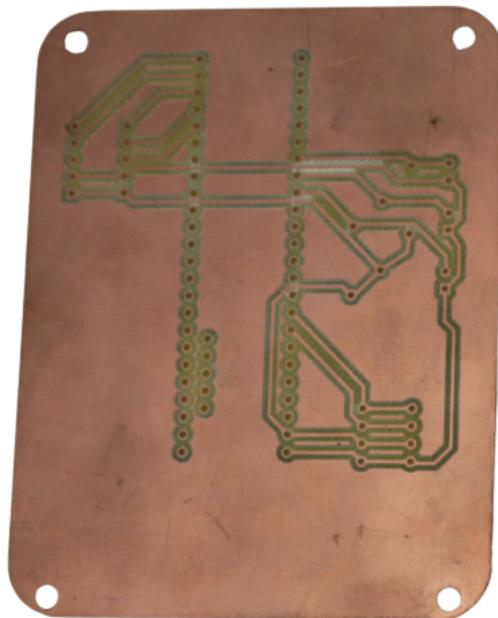
Una PCB o placa de circuito impresa por sus siglas en inglés, están compuestas de material aislante y pistas conductoras de cobre que logran interconectar diversos componentes electrónicos en una sola placa de metal. Este tipo de placas facilita la conexión entre componentes y optimiza el espacio para aplicaciones como lo es un instrumento musical electrónico.

Se diseñó la PCB de la OR-1 con la herramienta en línea EasyEDA y utilizando de referencia el esquemático de la sección anterior. Para poder conseguir un diseño modular de fácil montaje/desmontaje, se escogieron headers hembras para la conexión de los diversos componentes de la máquina a la placa. El diseño simulado se presenta en la siguiente imagen:



Puedes acudir al MakerLab de la Universidad del Valle de Guatemala para la impresión de tu PCB. En este laboratorio se cuenta con una máquina fresadora capaz de imprimir diseños en 2 capas. El diseño de PCB de la OR-1 cuenta con todos los requerimientos necesarios y ha sido aprobada por el encargado y supervisor del laboratorio, el Ingeniero Pablo Mazariegos.

Puedes acceder a los archivos Gerber (necesarios y listos para la impresión) directamente en el GitHub del proyecto. El resultado final es la PCB de la siguiente imagen:



5

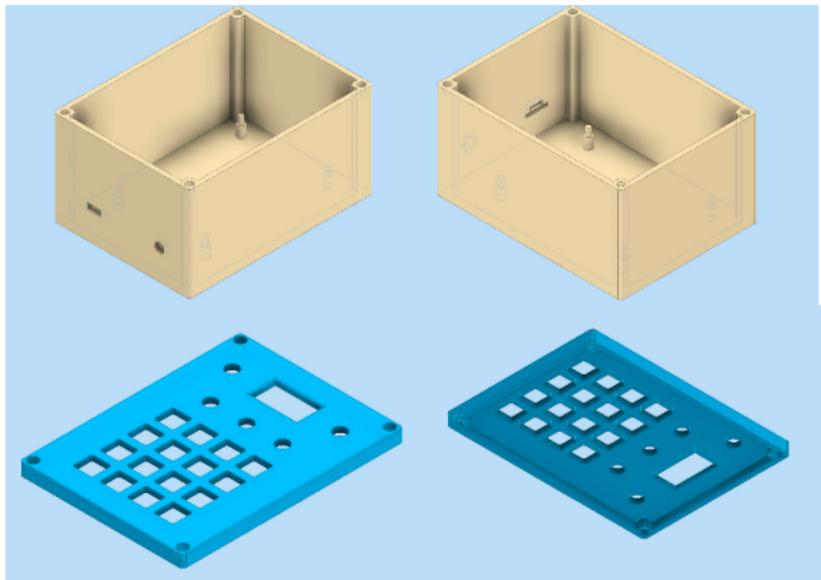
Diseño e impresión de la carcasa

La carcasa es una parte esencial de la OR-1 que permite la contención de todos los componentes, escondiendo aquellos que no son parte de la interfaz de usuario y garantizando el fácil acceso a periféricos tales como las salidas de audio y puerto de alimentación.

El diseño de la carcasa de la OR-1 se realizó utilizando la herramienta en línea OnShape, basándose en un diseño de caja con tapadera proporcionado por la empresa Lightning Boxes. Se añadieron los agujeros pertinentes para colocar de manera efectiva los botones, encoders, pantalla y el NeoTrellis. También se colocaron agujeros para el jack de audio y micro USB hembra. Finalmente, se agregó un soporte para la PCB, asegurando que esta no se mueva durante el proceso de ensamblaje ni al transportar el dispositivo de un lugar a otro.

Finalmente, la caja cuenta con agujeros para poder añadir insertos que permiten cerrar la caja con tornillos.

El diseño final de la caja se puede observar desde distintos ángulos en las siguientes imágenes:



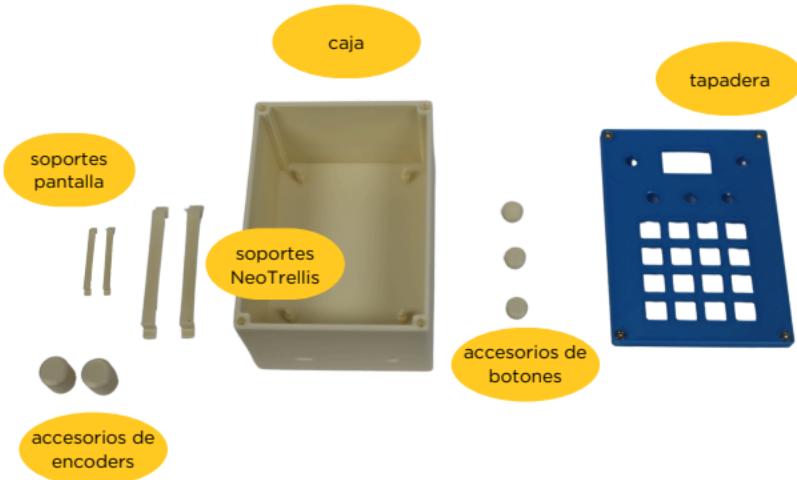
Todos los archivos STL del diseño pueden ser accedidos en el GitHub del proyecto. Puedes acudir al D-Hive en la Universidad del Valle de Guatemala para recibir una capacitación acerca de la impresión 3D y tener acceso a las impresoras para realizar tu proyecto.

El proceso de impresión requiere que descargas los archivos STL del proyecto y los transfieras a un programa slicer que utilice la impresora 3D disponible. Este lo aprenderás a utilizar durante la capacitación.

Se recomienda utilizar PLA, un material con propiedades biodegradables y de bajo costo y realizar la impresión a una altura de capa de 0.2mm, buscando el equilibrio entre resolución y velocidad de impresión.

Adicional a la caja, puedes descargar e imprimir accesorios para los encoders y botones, lo cual brinda cohesividad al diseño del instrumento. También debes imprimir los soportes para evitar que se mueva la pantalla y el NeoTrellis.

Las piezas finales impresas pueden observarse en la siguiente imagen:



6

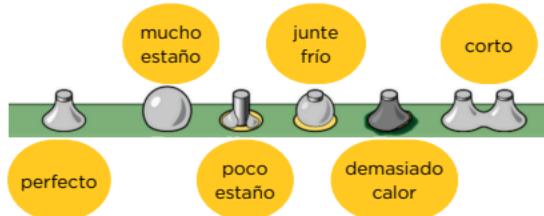
Soldadura

La soldadura es un proceso de fabricación que permite unir dos o más materiales para ensamblarlos. Para este proceso es necesario contar con un cautín de 20-60 Watts y punta de cincel, un soporte para cautín con inclinación al suelo y almohadilla, estaño, pasta para soldar y alcohol isopropílico. Todos estas herramientas pueden ser accedidas con la autorización correspondiente en los laboratorios de electrónica de la Universidad del Valle de Guatemala.

El proceso de soldadura puede resumirse en 4 pasos:

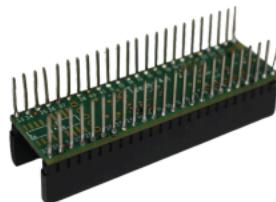


Es necesario limpiar la punta en la esponja mojada y aplicar una pequeña capa de estaño al inicio y entre conexiones.

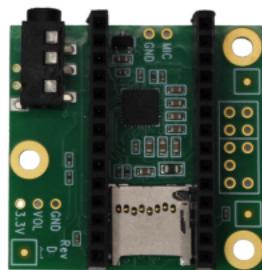


Para la construcción de la OR-1, deberás soldar lo siguiente:

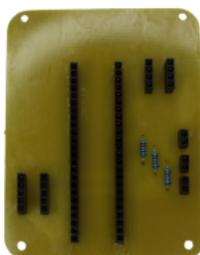
Headers hembra de pin largo al Teensy 4.1



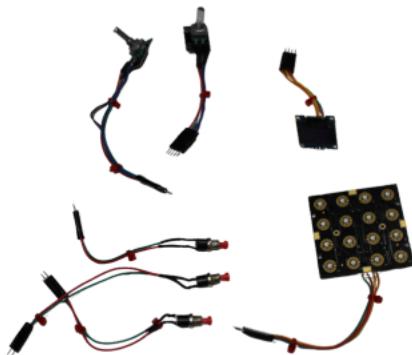
Headers hembra de pin largo al Audio Board Rev D



Headers hembra de pin corto a PCB



Cables dupont a encoders, botones, pantalla y NeoTrellis
utilizando código de color



Cable micro USB a PCB de micro USB hembra



Salida de audio estéreo del Audio Board a jack de 3.5mm



***Puede que no consigas la soldadura perfecta en el primer intento; pero con la práctica mejorará**

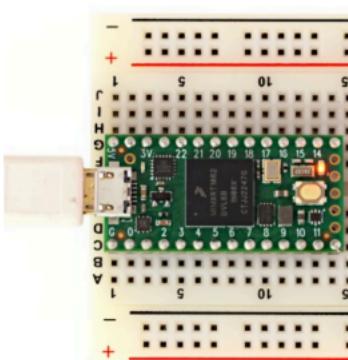
7

Programación y Código fuente

La programación de la OR-1 se realizó utilizando la versión 2.3.2 del Arduino IDE y el lenguaje de programación del mismo entorno. Antes de cargar el código fuente a tu Teensy, deberás realizar las siguientes configuraciones:

1. Verificación inicial del Hardware

Conecta tu Teensy para asegurarte que funcione. Colócalo en un protoboard para asegurar estabilidad y conéctala a tu computadora con el cable USB para realizar el test de la LED. Un Teensy nuevo debería tener un programa que hace parpadear el LED naranja lentamente. Si no parpadea, presiona el botón para ponerlo en modo de programación. El LED rojo debería encenderse indicando el modo de programación.



2. Instalación del Software Arduino IDE

Descarga e instalar la última versión del Arduino IDE (2.3.2) desde el sitio web oficial de Arduino en <https://www.arduino.cc/en/software>.

- a. Windows: ejecuta el instalador y sigue los pasos.
- b. MacOS: copia a aplicaciones y ejecuta el archivo instalador. Permite acceso a Documentos si es solicitado.
- c. Linux: extrae el archivo ZIP y ejecuta el programa.

3. Añadir el URL de Teensy en las preferencias

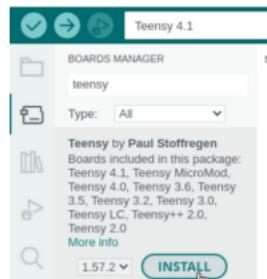
Al tener el Arduino IDE instalado, selecciona Archivo > Preferencias. En “Additional boards manager” añade:

https://www.pjrc.com/teensy/package_teensy_index.json



4. Instala Teensy en Boards Manager

En la pantalla principal de Arduino, abre Boards Manager y busca “Teensy” y haz click en “Instalar”



5. Instala Teensyduino

Descarga Teensyduino, un complemento para el Arduino IDE que agrega soporte para los microcontroladores Teensy, desde la página oficial de Teensy en:

https://www.pjrc.com/teensy/td_download.html

Arduino 1.8.x Software Development

Teensyduino is a software add-on for the Arduino software.

- [Macintosh Complete Software](#)
Supports: Catalina, Big Sur, Monterey, Ventura, Sonoma
- [Macintosh Installer for Arduino on Older Macs](#)
Supports: Mojave
- [Linux Installer \(X86 32 bit\)](#)
- [Linux Installer \(X86 64 bit\)](#)
- [Linux Installer \(ARM 32 bit / Raspberry Pi\)](#)
- [Linux Installer \(ARM 64 bit / AARCH64 / Jetson TX2\)](#)
- [Windows XP / 7 / 8 / 10 / 11 Installer](#)

Uso del Teensy Loader:

El teensy loader es una herramienta que se instala automáticamente con Teensyduino y se ejecuta de fondo cuando se programa el Teensy desde el Arduino IDE.



Librerías

Deberás descargar las siguientes librerías:

- Adafruit_NeoTrellis (V1.3.3) de Adafruit
- Encoder (V1.4.4) de Paul Stoffregen
- Audio (nativa del Teensy AudioBoard)
- Wire y SPI
- SerialFlash
- Adafruit_GFX (V1.11.9)
- Adafruit_SSD1306 (V2.5.10)

Estas librerías fueron utilizadas para la programación del dispositivo y son esenciales para su funcionamiento.

Programación y monitoreo:

Luego de descargar el código fuente desde el GitHub del proyecto y abrirlo en Arduino IDE, compílalo presionando el botón “verificar” (cheque) y luego “cargar” (flecha) en el Arduino IDE.



Puedes monitorear las salidas con el monitor serial.



Código fuente

El código fuente cuenta con distintas funciones que cumplen diversos propósitos en el funcionamiento de la OR-1. A continuación, se brinda una pequeña explicación de las funciones más importantes. Se recomienda leer detalladamente el código para comprender cómo se logra llegar a la función que realiza cada una.

1. Funciones de Inicialización y configuración:

- *setup()*: configura la comunicación serial, inicializa el hardware y las conexiones de audio.
- *initializeHardware()*: inicializa componentes del hardware como la pantalla SSD1306 y el Trellis, establece los pines de entrada y realiza la animación de inicio.
- *initializeAudioConnections()*: configura los objetos de audio y conexiones entre ellos, incluyendo mezcladores y efectos.
- *assignColors()*: asigna colores a cada pad del Trellis para una mejor distinción visual.

2. Funciones de manejo de la interfaz de usuario

- *toggleProgramMode()*: cambia entre el modo de programación y el modo de reproducción libre.
- *toggleStartStop()*: inicia o detiene la secuencia de reproducción de los sonidos. *handleBpmAdjustment()*: ajusta el BPM (beats por minuto) basado en la entrada del encoder principal.
- *toggleFxAdjustmentMode()*: activa o desactiva el modo de ajuste de efectos basado en el estado del botón de efectos.
- *handleEffectsEncoder()*: maneja la selección y ajuste de efectos usando el encoder de efectos.

3. Funciones de procesamiento y control de Audio

- *updateSequencer()*: actualiza el secuenciador que controla que muestra de sonido se reproduce en cada paso.
- *handleRepeat()*: maneja la repetición de un sonido específico si se activa el modo de repetición. *adjustVolume()*: ajusta el volumen general del sistema.
- *adjustHighPassFreq()*: ajusta la frecuencia del filtro pasa altos.
- *adjustGranularFx()*: modifica los parámetros del efecto granular.
- *adjustFxParameter()*: ajusta los parámetros del efecto activo actualmente basado en la entrada del usuario.

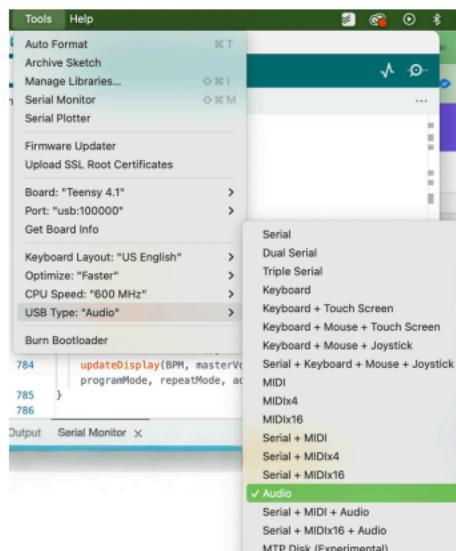
4. Funciones de ayuda y visualización

- *startupAnimation()*: realiza una animación visual en el Trellis al iniciar.
- *printArray()*: imprime el estado actual del secuenciador para propósitos de depuración.
- *displayMenu()*: muestra el menú de efectos en la pantalla.
- *updateDisplay()*: actualiza la pantalla OLED con información como BPM, volumen y estado de los modos.
- *DrawDial()*: dibuja un dial en la pantalla para representar valores como volumen o parámetros de efectos.

Los *samples* utilizados durante la programación de la OR-1 son parte de un kit gratuito de drumkit, con los sonidos originales de la TR-808 de Roland Corporation. Se escogieron 16 y se cargaron a la micro SD. Puedes descargarlos en el GitHub del proyecto, o si deseas utilizar otros, puedes cargarlos a tu tarjeta y modificar los nombres en el código fuente.

Configuración de salidas de audio

El Teensy inicialmente será configurado como un dispositivo serial con salida de monitoreo por medio del jack para audífonos. Para poder utilizar el puerto micro USB como interfaz y aprovechar la máxima calidad de audio , deberás re-configurarlo de la siguiente manera en el Arduino IDE: herramientas > USB Type > Audio



8

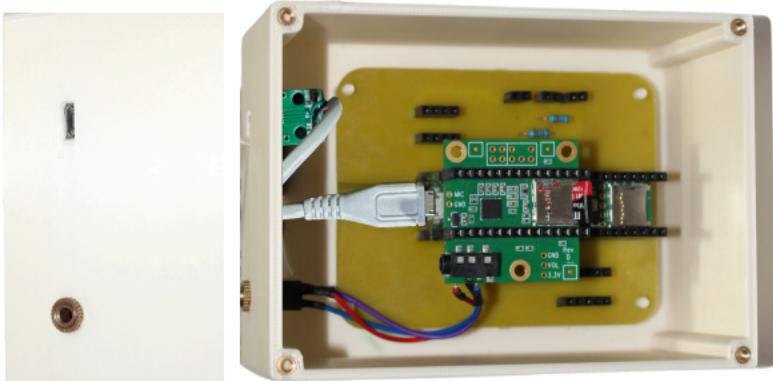
Ensamblaje

Antes de continuar con el ensamblaje de la OR-1, es necesario que hayas completado todas las secciones anteriores, incluyendo la impresión 3D de la carcasa y haber realizado pruebas de la funcionalidad del circuito PCB utilizando los componentes con sus cables soldados.

Al tener todo listo y los samples cargados en la micro SD del AudioBoard, podemos proceder a montar todo en la carcasa en 4 pasos:

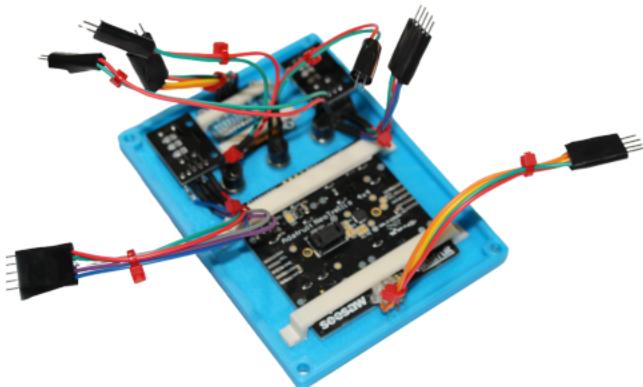
1

Colocar el Teensy en su header correspondiente del PCB e introducirlo a la caja. Luego, enroscar la salida de audio en su agujero correspondiente y montar el AudioBoard sobre el Teensy. Posteriormente, conectar el cable micro USB con el PCB hembra y colocarlo en su agujero como se muestra en la imagen.



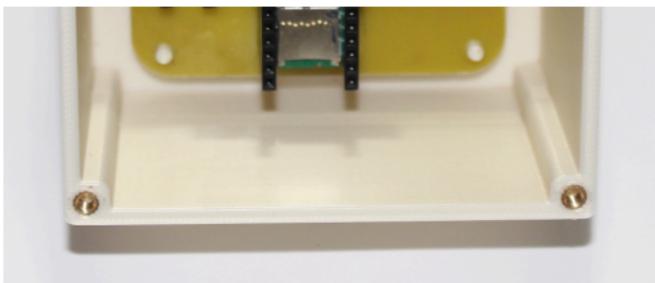
2

Colocar los componentes de la interfaz de usuario (botones, encoders, NeoTrellis y pantalla) en la tapadera de la caja, junto con los soportes necesarios para los dispositivos I2C.



3

Con la ayuda de un cautín, colocar los insertos de rosca en las cuatro esquinas de la caja



4

Con mucho cuidado y utilizando como referencia el esquemático, conectar los dispositivos de la tapadera a sus headers designados en el PCB y asegurar la caja utilizando tornillos en las esquinas. Finalmente, añadir los accesorios de los encoders y botones.



9

Ejercicios sugeridos

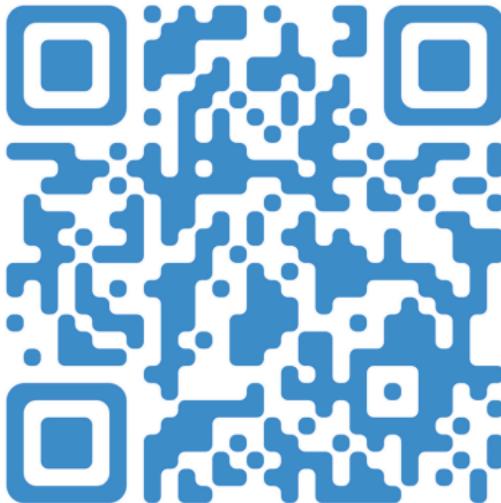
La OR-1 es una máquina con una capacidad relativamente limitada, y la idea de crear tu propio instrumento es poder personalizarlo acorde a tus necesidades. Para esto, se sugiere que explores con el código fuente y crees nuevas funciones que ayuden a potenciar tu creatividad musical. Se proponen los siguientes ejercicios:

- Crea un nuevo efecto/filtro que pueda ser controlado utilizando el knob de efectos:
 - Analiza el código fuente y la cadena de audio que va desde las samples a la salida stereo y USB, agregando la nueva funcionalidad a la misma y logrando su reproducción sin afectar el resto de efectos.
 - Agrega el nombre del efecto/filtro al menú de visualización en la pantalla.
 - Asegúrate que el efecto/filtro creado no cause distorsión en la salida al tener el volumen de la OR-1 en su máxima potencia.
- Instrumento MIDI
 - Investiga en la documentación de Teensy en línea si es posible convertir la salida de audio USB a salida MIDI.
 - Modifica el código fuente para cambiar la salida a MIDI.

- Agrega la capacidad de reproducir patrones pre creados o guardar patrones directamente desde la máquina.
 - Programa secuencias en el código fuente para que puedan ser seleccionadas utilizando el knob de efectos o alguno de los botones.
 - Añade la función de poder guardar el patrón que programaste con el secuenciador para una futura reproducción del mismo.
- Optimiza la pantalla de visualización con animaciones que le agreguen un toque diferente al instrumento.
 - Mientras no se modifica ningún parámetro, agrega una animación de una persona bailando. Este debe pausarse si se pausa la secuencia.

No existe un límite en cuanto a las cosas que se pueden hacer con la OR-1. Su diseño *Open Source* y la capacidad de personalización de la misma busca que puedas explorar, innovar y desarrollar tus propias aplicaciones creativas.

10 GitHub



<https://github.com/andreefuentes/OR1>



mayo 2024