

Programación avanzada
Instrumentación Electrónica
José Alfonso Domínguez Chávez

Pedal con Arduino
**Segunda revisión de avances
del proyecto final**

Josué Marcelo Castillo Acosta

Kaylee Michelle Díaz

Rodríguez

Juliana Hernández Hernández

Hugo Jesús Navarro Hernández

César David Salas Martínez



Introducción:

Palabras clave: LTspice, Fuzz, .WAV, Audacity, onda, sonido, Python, GeoGebra, Arduino IDE, Microcontroladores, Arduino UNO, ESP32, Github

En este segundo avance tenemos que entregar la mayor parte del sistema ya funcional, para así poder localizar cualquier error antes de la entrega final, aquí mostraremos todos los avances que se han realizado, junto a pruebas de funcionamiento, investigaciones realizadas, códigos ocupados, y simulaciones hechas con LTspice.

Todo este contenido se puede encontrar en el repositorio de [Equipo-NANA/Proyecto at main · Choclotherock/Equipo-NANA](#)

En el apartado de proyecto, también debajo de menciones de archivos se pondrá un enlace y el nombre del archivo del repositorio.

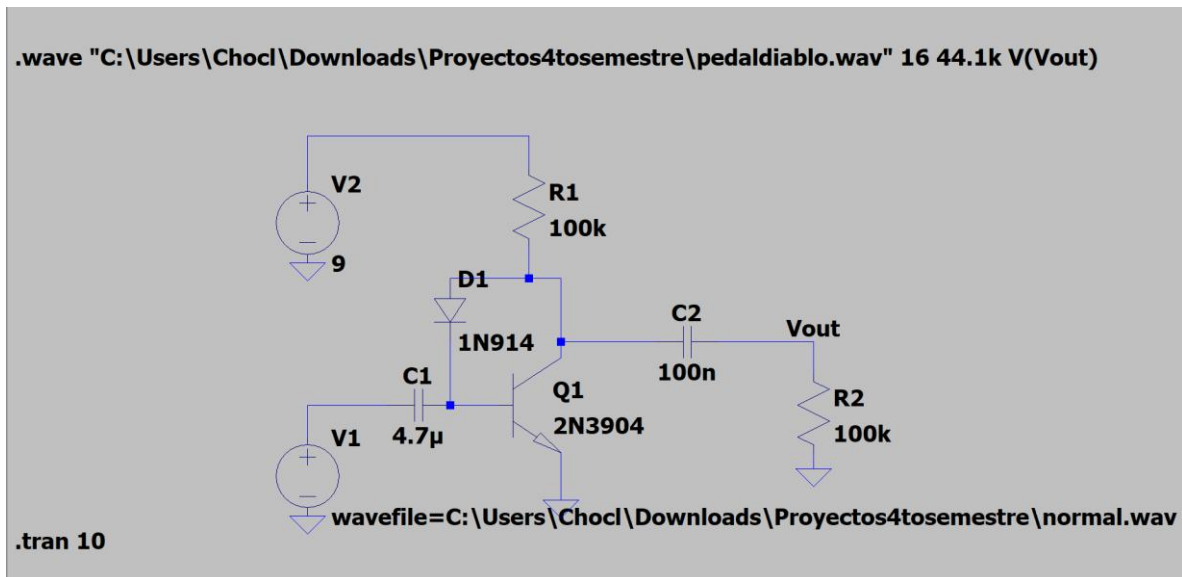
1.-Sistema funcional:

1.1.-Sonido modificado:

Aquí estará la investigación que se realizó sobre la comparativa entre un sonido normal de un bajo eléctrico, y el mismo sonido puesto a través de un pedal de fuzz tradicional, para poder saber como es que el sonido se modifica a partir de este circuito.

Para esto primero se utilizó una nota ya grabada de un bajo eléctrico para luego pasarla por una simulación en LTspice, donde estaba un circuito de pedal de fuzz genérico, para que después la salida de este circuito se guardara como archivo .WAV, donde se puede reproducir y analizar su onda con el software Audacity.

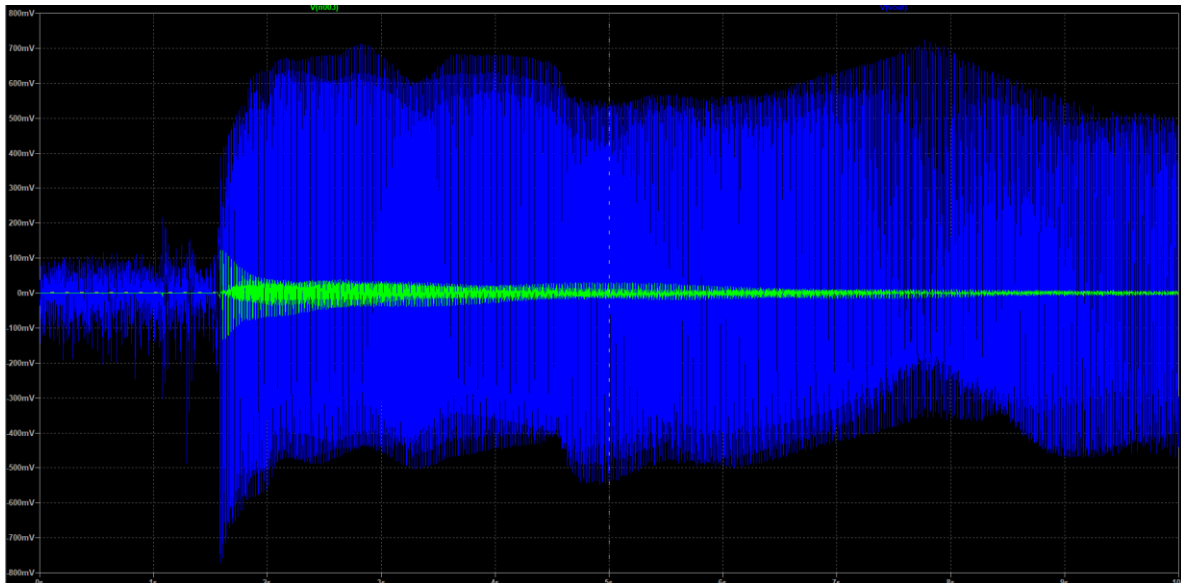
Aquí el circuito realizado para esto:



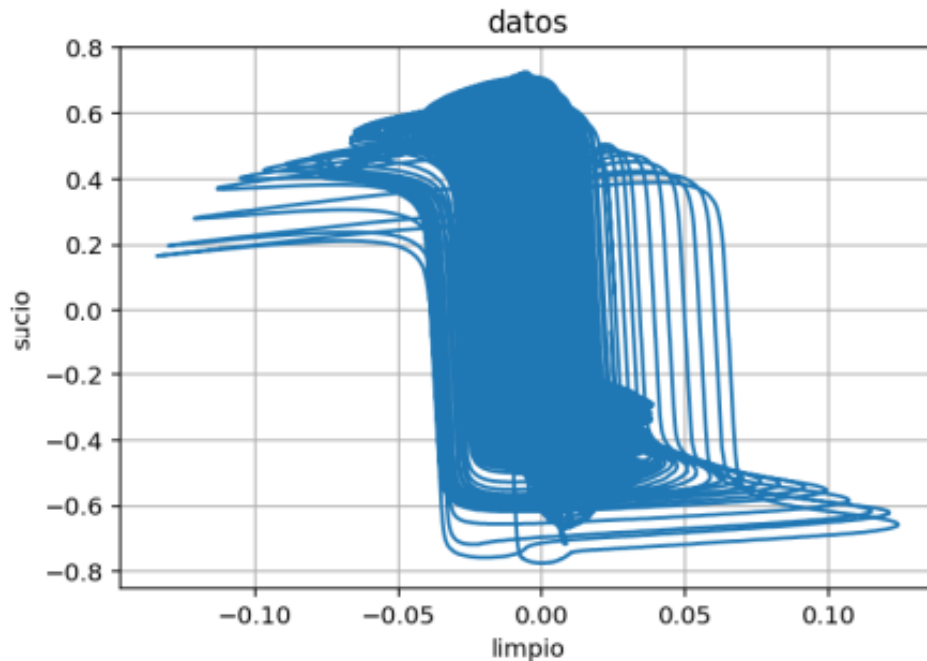
[Equipo-NANA/Proyecto/SIMULACIONES at main · Choclotherock/Equipo-NANA](#)

-Simulacion de pedal fuzz PARA PRUEBAS

Ya al realizar esto y poder correr la simulación, tenemos los dos archivos de onda para poder analizar, usamos un análisis transitorio de 10s para esto, aquí los resultados:



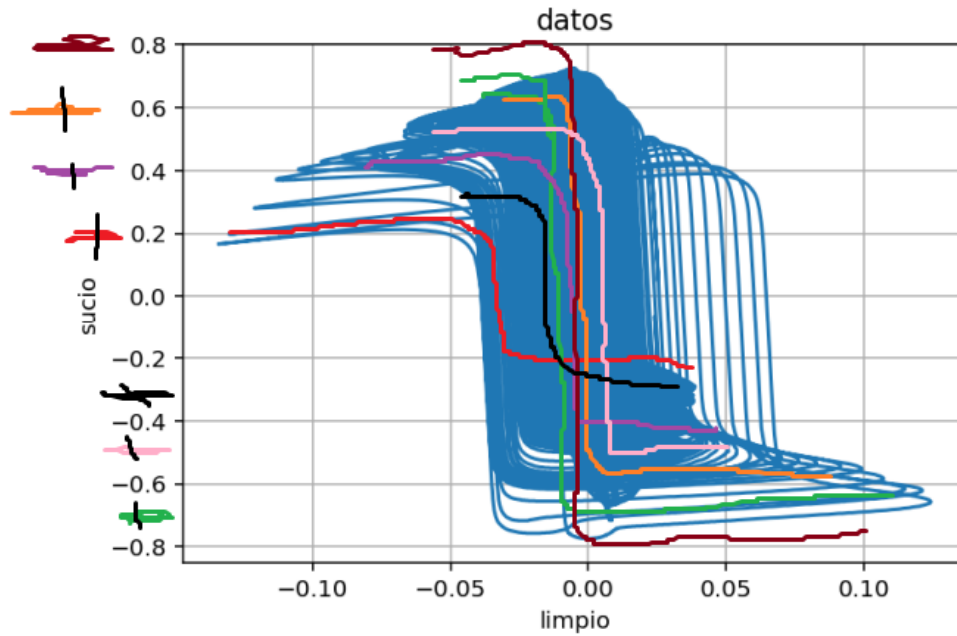
Como podemos ver, el circuito de fuzz hace que se amplie en gran medida la onda, pero no parece ser una amplitud lineal, es por esto que guardamos estas dos ondas como datos en un archivo .txt, el cual pasaremos por un código de Python, para poder graficar los datos de entrada contra los de salida, siendo el eje X la amplitud de la entrada del audio mientras Y es la amplitud de la salida del audio, obteniendo un resultado así con el código:



[Equipo-NANA/Proyecto/CODIGOS at main · Choclotherock/Equipo-NANA](#)

-Codigo para graficacion de comparativa de entrada y salida

Con esta graficación tenemos la comparativa entre la entrada y salida, como podemos ver esta, “amplificación” no es lineal, si no que se parece mas a una función logística, conocida también como una función con forma de S, ya con esto visto se realiza una investigación sobre como son las formulas de estas ecuaciones, tomando en total 7 ecuaciones que se reflejan en las siguientes partes de esta grafica:



Aquí las ecuaciones y su grafiación en GeoGebra

$$f1(x) = \frac{0.4}{1 + e^{100(x+0.03)}} - 0.2$$

$$f2(x) = \frac{0.6}{1 + e^{90(x+0.02)}} - 0.3$$

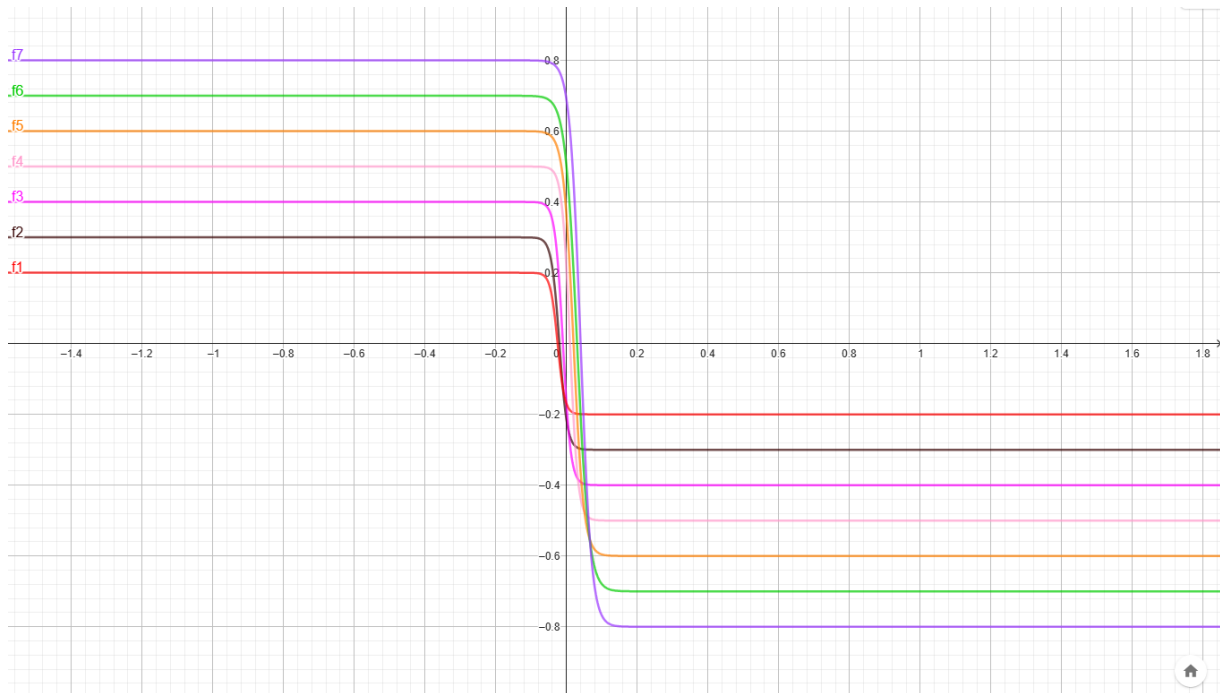
$$f3(x) = \frac{0.8}{1 + e^{90(x+0.01)}} - 0.4$$

$$f4(x) = \frac{1}{1 + e^{90(x-0.01)}} - 0.5$$

$$f5(x) = \frac{1.2}{1 + e^{70(x-0.02)}} - 0.6$$

$$f6(x) = \frac{1.4}{1 + e^{60(x-0.03)}} - 0.7$$

$$f7(x) = \frac{1.6}{1 + e^{65(x-0.04)}} - 0.8$$



[Equipo-NANA/Proyecto/SIMULACIONES at main · Choclotherock/Equipo-NANA](#)

-Ondas para el pedal

Con estas ecuaciones, en teoría podríamos replicar el efecto del Fuzz haciendo que los datos de entrada de audio sean utilizados como X en estas ecuaciones para luego poder sacarlas afuera del sistema, esto siendo realizado mas adelante en este trabajo.

Codigo usado para esto:

[Equipo-NANA/Proyecto/CODIGOS at main · Choclotherock/Equipo-NANA](#)

-Ecuaciones codificadas

1.2.-Pantalla OLED

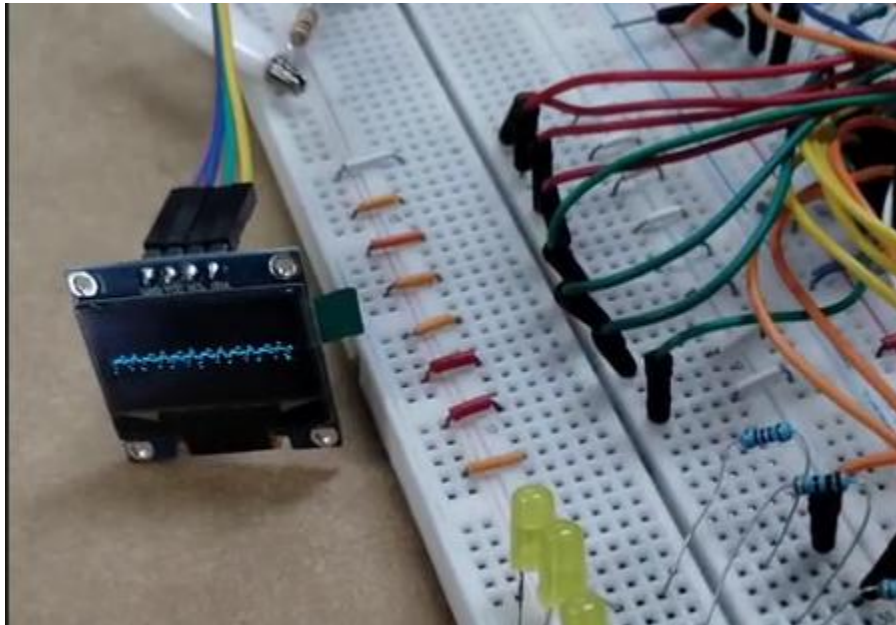
Para esta parte del trabajo se necesita el poder saber que librerías usar para su correcto funcionamiento, a parte de saber como enviarle los datos correctamente, para esto se uso la pantalla la cual se mostro en el anterior trabajo, se utilizo el ArduinoIDE, donde se utilizaron las librerías

-Wire.h

-Adafruit_GFX.h

-Adafruit_SSD1306.h

Ya con estas librerías podemos empezar a graficar la onda con la que contamos, esto lo hacemos con nuestro código y aquí un poco de como se ve ya funcionando,



[Equipo-NANA/Proyecto/CODIGOS at main · Choclotherock/Equipo-NANA](#)

-OLED

[Equipo-NANA/Proyecto/VIDEOS at main · Choclotherock/Equipo-NANA](#)

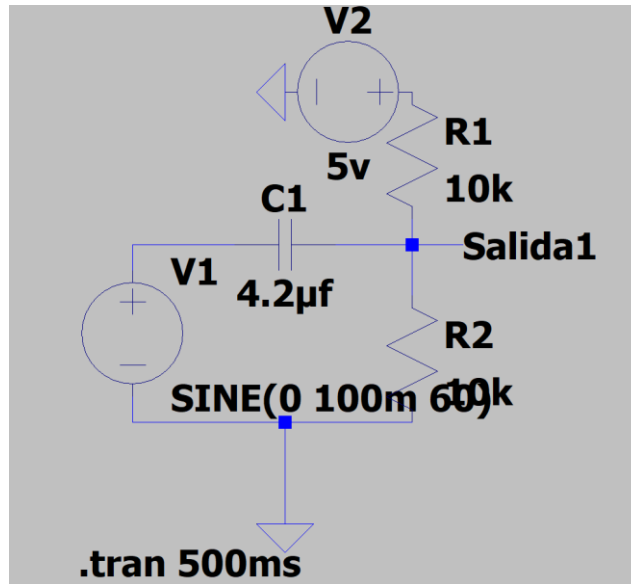
-Pantalla OLED prueba

Ya con esta graficacion ya podríamos en teoría agregarlo al demás código para graficar las ondas modificadas, solamente modificando un poco este código base.

1.3.-Entrada de datos:

En una de las partes mas importantes de todo este proyecto es la entrada de datos al microcontrolador, ya que los datos de un instrumento electrico se suelen generar por las llamadas “Pastillas”, estas lo que hacen es captar con imanes las vibraciones de las cuerdas metálicas, esto después pasa a ser un voltaje eléctrico que ya sale del instrumento hacia un amplificador debido al tamaño de los datos.

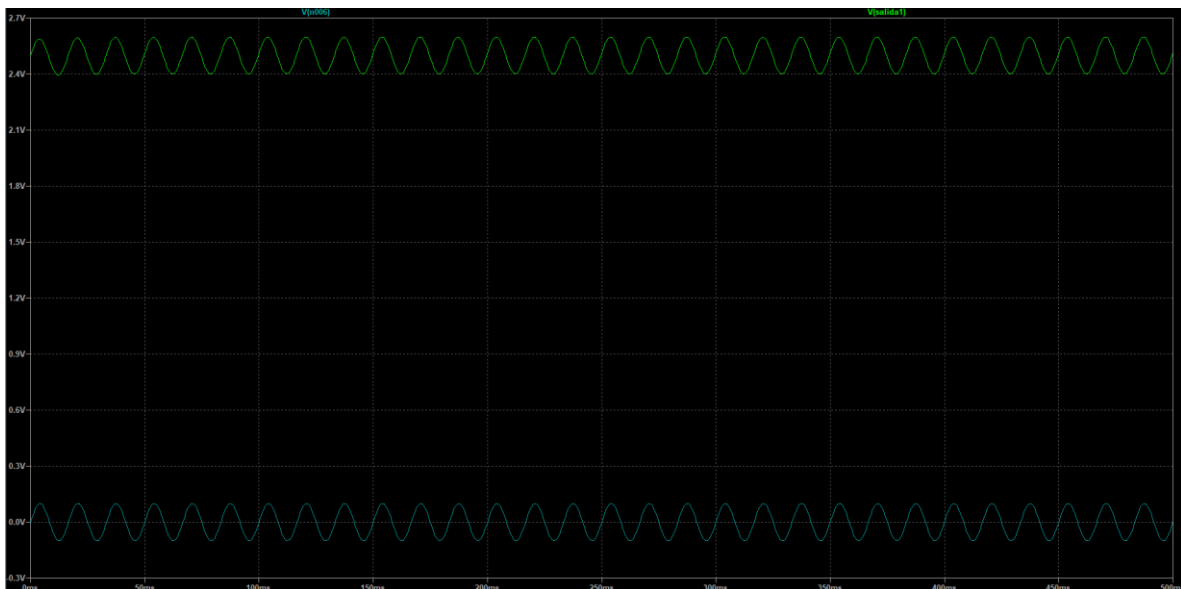
Entonces para esto primero necesitamos ver como son los datos naturalmente, los cuales oscilan alrededor de 0v y bajan aproximadamente entre 120mV y -120mV, esto siendo un problema, ya que los microcontroladores solamente pueden recibir voltaje positivo, entonces primero tendremos que hacer un circuito de offset, con el cual vamos a subir los datos una medida, primero realizamos la simulación para comprobar su correcto funcionamiento.



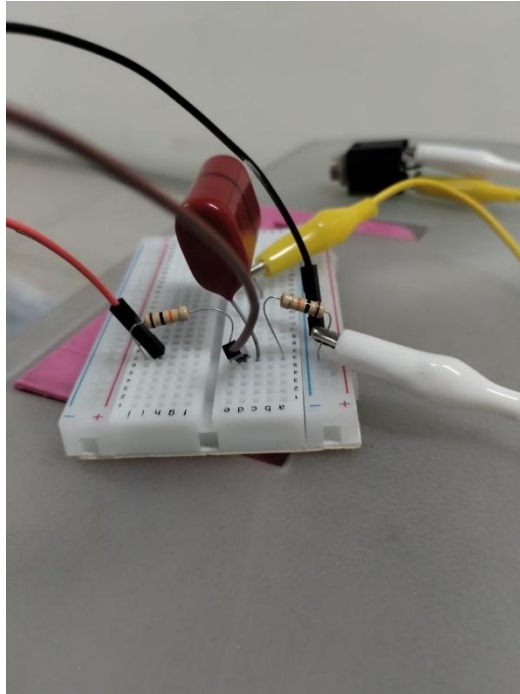
[Equipo-NANA/Proyecto/SIMULACIONES at main · Choclotherock/Equipo-NANA](#)

-Simulaciones para la entrada y salida del audio con Amplificadores operacionales

Aquí los resultados obtenidos:



Como podemos notar el circuito funciona debidamente, así que armando el mismo circuito en la vida real realizamos la entrada de datos.



Ya con los datos adentro del microcontrolador podemos empezar a checar que realizar con estos mismos.

1.4.-Análisis de datos:

Para poder analizar los datos correspondientes, primero podemos solamente ver los datos de entrada, graficarlos en el Arduino IDE para poder ver si están entrando de buena manera al sistema, y para comprobar la calidad de este audio primero se empezaron a hacer pruebas de mandar los datos a un archivo de audio con un código de Python, con resultados mixtos, ya que al mandar los datos y escucharlos, nos dimos cuenta que al usar el Arduino UNO, este por las características de su entrada analógica, cortaba y distorsionaba bastante el audio mandado, probablemente por su velocidad de entrada de datos, es por esto que se decidió cambiar al ESP32, ya que este cuenta con una mayor velocidad de lectura y un mayor número de bits para lectura analógica, además de contar con una salida DAC que nos será de gran ayuda en la etapa de salida del audio, ya con esto se empieza a ocupar el ESP32 como microcontrolador para este proyecto, se empieza a realizar esta entrada y salida de datos en limpio, obteniendo resultados más agradables como el siguiente audio visto en Audacity



[Equipo-NANA/Proyecto/PRUEBASDIGITALES at main · Choclotherock/Equipo-NANA](#)

-Grabacion28

Ya con esto, se modifica el código para que implemente las diferentes ecuaciones para ver cual le acomoda mejor al audio, aquí esta una de las pruebas con una de las anteriores ecuaciones,



[Equipo-NANA/Proyecto/PRUEBASDIGITALES at main · Choclotherock/Equipo-NANA](#)

-Grabacion 29

Ya visualmente se puede ver la modificación de esta onda con el código propuesto, describiendo el sonido como mas agudo que el original y mas fuerte, ya con esta base se basaran las siguientes modificaciones deseadas.

Todas las pruebas de audio realizadas aquí

[Equipo-NANA/Proyecto/PRUEBASDIGITALES at main · Choclotherock/Equipo-NANA](#)

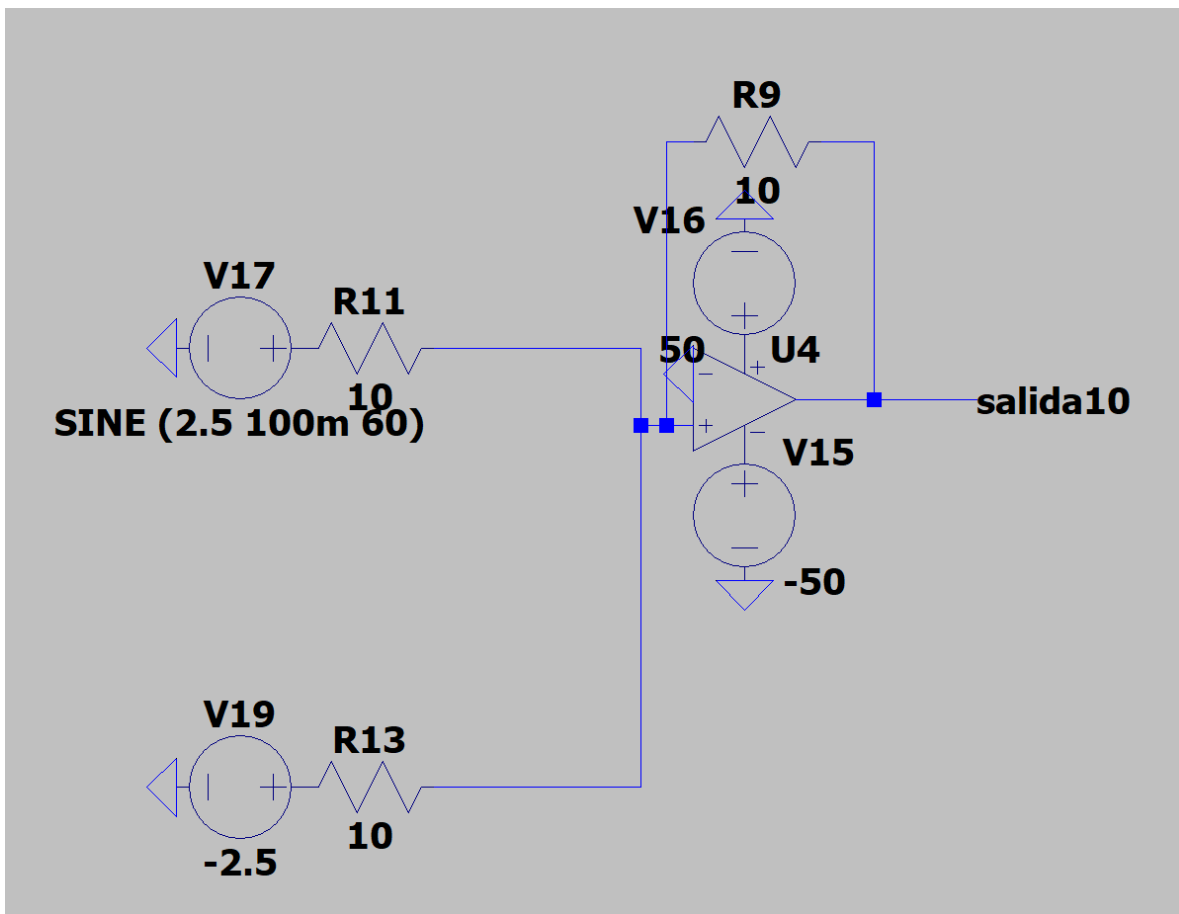
Codigos para esta parte

[Equipo-NANA/Proyecto/CODIGOS at main · Choclotherock/Equipo-NANA](#)

- LecturaLimpia
- EfectoPedalPrueba1
- Guardado de datos en WAV

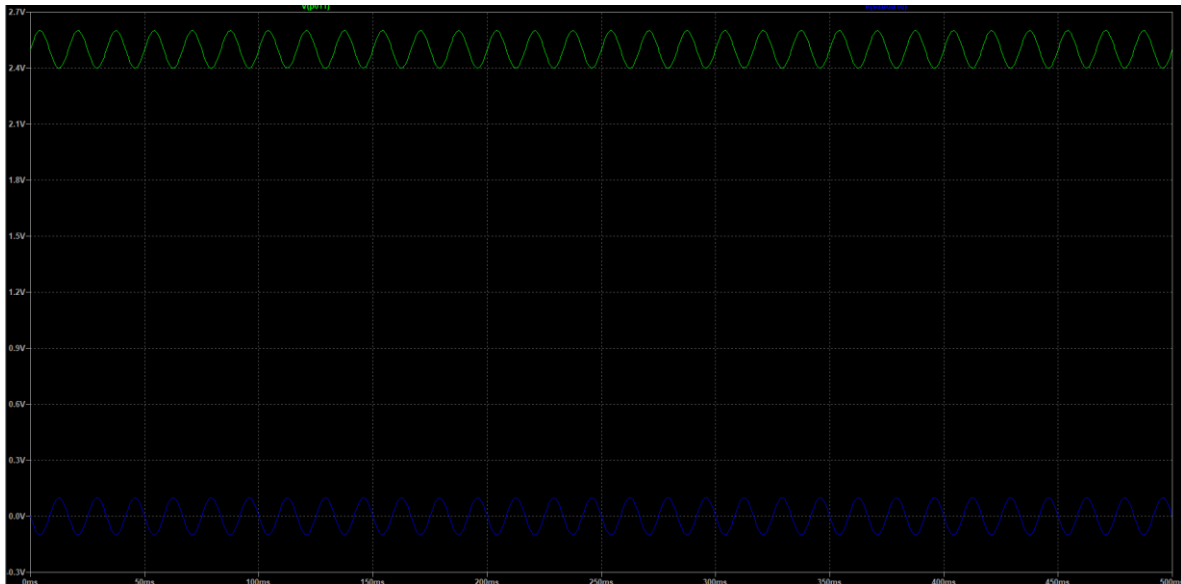
1.5.-Salida de datos:

Para la salida de estos datos a un amplificador de guitarra estándar, se tienen que realizar varios pasos distintos, que tiene que ver con la naturaleza más física de este proyecto, ya que se necesitan sacar estos datos del ESP32 y mandarlos al amplificador, esto teniendo el problema de que no podemos sacar datos negativos de su salida DAC, entonces tendremos que sacar los datos de manera totalmente positiva, con el offset ahora de 1.650V, para primero pasarlos a un circuito sumador, el cual por sus propiedades también invertirá la señal, entonces se tendrá que volver a invertir con un circuito inversor, estos 2 usando un amplificador operacional TL072CP, aquí las simulaciones realizadas.

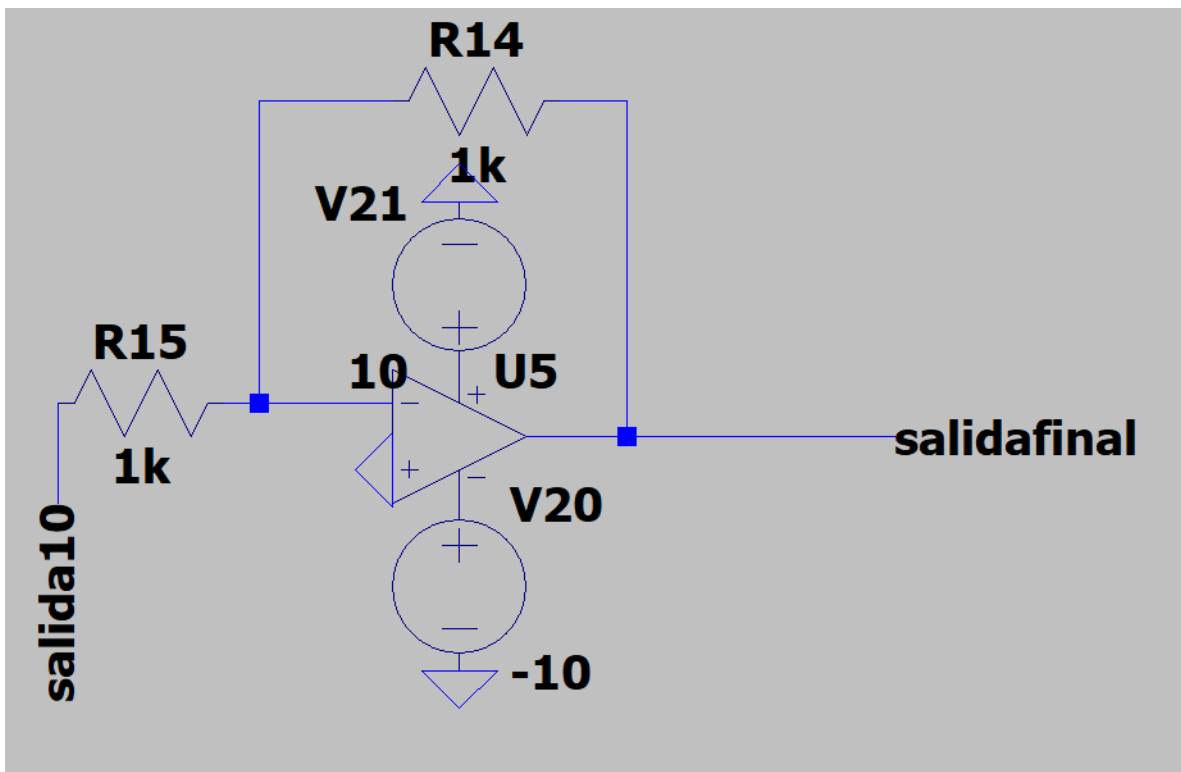


[Equipo-NANA/Proyecto/SIMULACIONES at main · Choclotherock/Equipo-NANA](#)

-Simulaciones para la entrada y salida del audio con Amplificadores operacionales

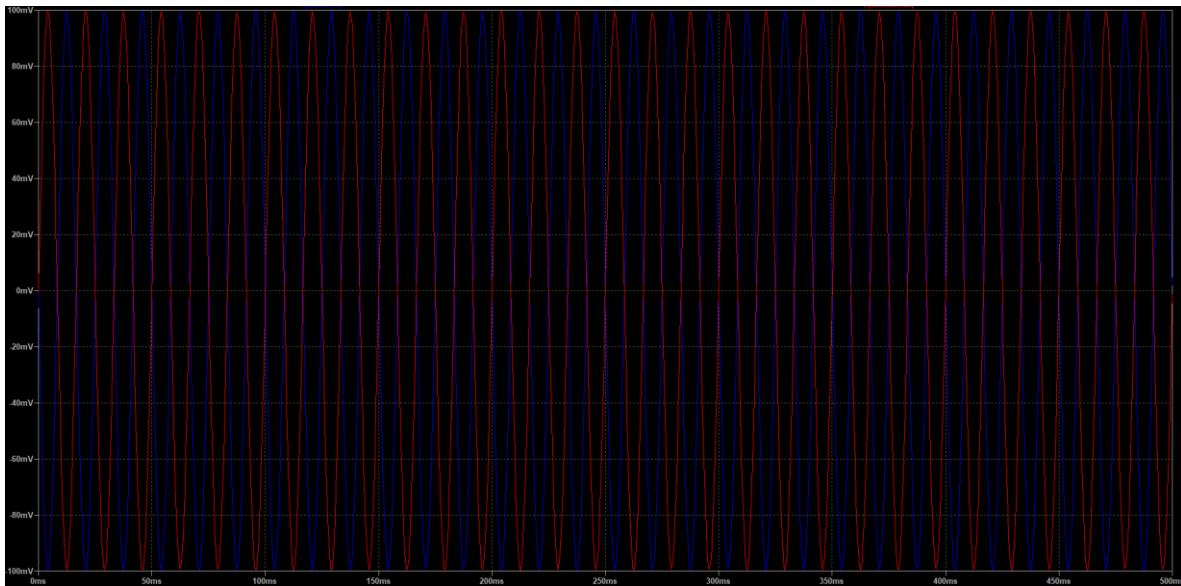


Circuito sumador inversor



[Equipo-NANA/Proyecto/SIMULACIONES at main · Choclotherock/Equipo-NANA](#)

-Simulaciones para la entrada y salida del audio con Amplificadores operacionales

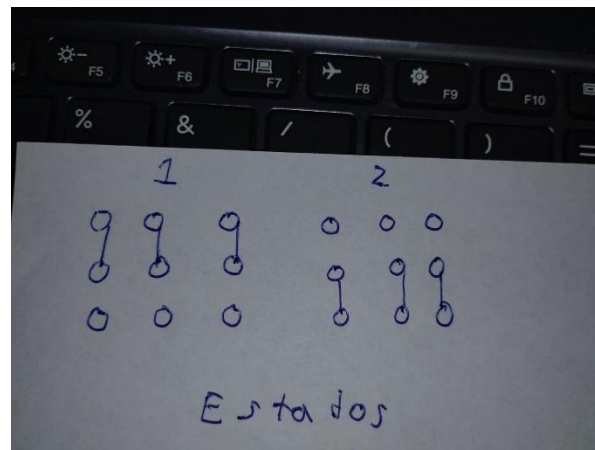


Circuito inversor.

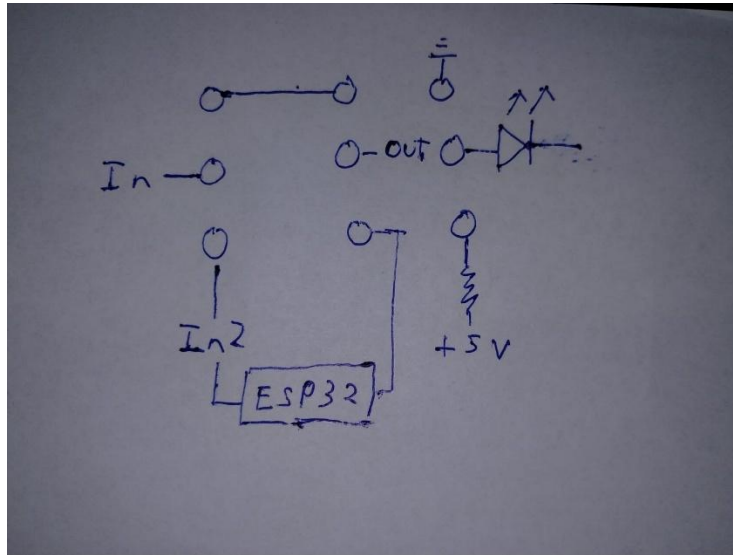
Ya así obteniendo una salida que oscila alrededor de 0 y que no este invertida, conservando de la mejor manera los datos.

1.6.-Cricuito de apagado y encendido

Este necesita para poder interactuar con el usuario un botón que prenda y apague el efecto así que se va a usar un botón de enganche de 9 pines, el cual puede variar entre 2 estados, el cual de va a conectar de la siguiente manera para su correcto funcionamiento:



El botón siempre estará en uno de los dos estados, solamente cuando se presione cambiara de estado y se quedara en ese hasta que se vuelva a presionar.



Este diagrama marca solamente las conexiones externas del botón, sin marcar ningún estado puesto para evitar confusiones, siendo “IN” la entrada del audio, “IN2” la entrada al ESP32 y “OUT” la salida del audio.

2.-Interfaz de usuario:

El usuario va a interactuar con el pedal únicamente teniendo que conectarlo a la corriente eléctrica y de ahí presionando el botón si quiere el efecto prendido u apagado, ahí en la interfaz pudiendo ver la modificación de onda y la luz del LED si esta prendido o apagado.

Teniendo que conectar su instrumento a la parte de entrada del circuito y conectar el amplificador a la parte de salida del circuito.

3.-Informe de fallas cosas pendientes y desempeño general.

3.1.-Fallas/Pendientes:

Como se puede ver hubo distintas fallas encontradas cuando se intento utilizar el Arduino UNO para el procesamiento de la señal, por culpa de sus características inferiores, pero la solución fue sencilla ya que también se cuenta con el ESP32 el cual se sigue programando con el Arduino IDE, manteniendo la mayor parte del código igual

[Equipo-NANA/Proyecto/PRUEBASDIGITALES at main · Choclotherock/Equipo-NANA](#)

-Grabaciones de la 1 a la 6

Otro problema también encontrado es la entrada del ESP32 al ser en 12 bits se tiene que elegir entre subir esta entrada de manera digital hasta 16 o bajarla a 8 esto para poder hacer que el archivo .WAV se guarde correctamente, bajarla a 8 tiene sus desventajas como perdida de calidad de audio, pero la subida a 16 no se pudo lograr de manera que no entrara un ruido excesivo, volviendo inutilizable el audio guardado, entonces se decidió utilizar estas pruebas con 8 bits, pero se podrían dejar los 12 bits cuando se tenga que sacar a un entorno físico en lugar de digital.

[Equipo-NANA/Proyecto/PRUEBASDIGITALES at main · Choclotherock/Equipo-NANA](#)

-Grabaciones de la 7 a la 21, junto a la 25, problemas con 16 bits

Queda como pendientes el juntar los códigos graficadores y el código de modificación de audio, así como encargarse de cambiar un poco el código de modificación de audio para poder hacer un efecto de mejor calidad, también el circuito físico de salida del audio ya modificado, esto se realizara pronto ya que se cuentan ya con los Amplificadores operacionales y los demás componentes para este circuito.

3.2.-Pruebas de desempeño:

Ya se cuentan con bastantes resultados, en su mayoría son pistas de audio o videos que se han tomado, todos disponibles en el repositorio de GitHub.

[Equipo-NANA/Proyecto/PRUEBASDIGITALES at main · Choclotherock/Equipo-NANA](#)

-Grabaciones

Pero como pudimos ver anteriormente, ya con el código actual se pudo obtener un audio bastante aceptable, este si contaba con ruido, pero no era tanto en comparación de otras pruebas también realizadas, y el audio ya modificado tenia buena calidad y el ruido con el que contaba no era muy molesto, así que se pueden considerar exitosas.

Solamente con el análisis de estas ondas, nos pudimos dar cuenta que el efecto estaba cortando gran parte de los armónicos mas graves, haciendo que el audio se escuchara de manera mas aguada, suponemos que esto es por el uso de una sola de las ecuaciones que teníamos, esto lo veremos mas adelante con el uso de otras ecuaciones diferentes.

La mayor falla fue la modificación del audio, pero como se dice se seguirán haciendo pruebas con diferentes ecuaciones y con diferentes formas de modificar este audio, cada una con diferentes propósitos, además de que la salida probablemente no sea necesaria bajarla de bits cuando ya se tenga el circuito físico, arreglando ese bajón de calidad que se tenia.

4.-Conclusion:

Con esto concluimos el avance, como se puede ver ya se tiene la modificación del audio de una manera digital, con esto ya solamente nos quedara el tener que sacar el audio a una salida tipo Jack, así pudiendo concluir el proyecto, o al menos su parte principal.

Cualquier archivo que no tenga enlace adjunto probablemente se encuentre en el repositorio:

[Choclotherock/Equipo-NANA: Pues es un equipo, que pues equipea y puas asi](#)

En el apartado de Proyecto.