SINTETIZADOR ARDUINO

TOMAS HURRELL, JUAN MANUEL BAEZ, NICOLAS FERNANDEZ

Julio 2019

1 Introducción

La idea es poder realizar un sintetizador de audio, con el cual se pueda jugar con distintos filtros, logrando distintos sonidos y poder tocar música. Se pensó el proyecto para que sea entendible para quien quiera leer el código fuente, reproducir el proyecto y poder expandir la funcionalidad del mismo, así como también poder expandir la cantidad de teclas, potenciómetros que manejen efectos, o cualquier otra funcionalidad que se le ocurra agregar.

2 Conceptos

2.1 Sintetizador

Un sintetizador es un instrumento musical de tipo electrónico que, a través de circuitos, genera señales eléctricas que luego son convertidas a sonidos audibles. Una característica que diferencia al sintetizador de otros instrumentos electrónicos es que sus sonidos pueden ser creados y modificados. Los sintetizadores pueden imitar otros instrumentos o generar nuevos timbres.

2.2 Oscilador

Oscilador de audio es el nombre coloquial por el que se conoce a los circuitos electrónicos que generan a su salida una tensión eléctrica que oscila con una frecuencia comprendida entre 20 Hz y 20000 Hz aproximadamente, frecuencias que establecen los límites de audición de señales acústicas por parte del oído humano. Un oscilador electrónico es un circuito electrónico que produce una señal electrónica repetitiva, a menudo una onda senoidal o una onda cuadrada.

2.3 Mozzi

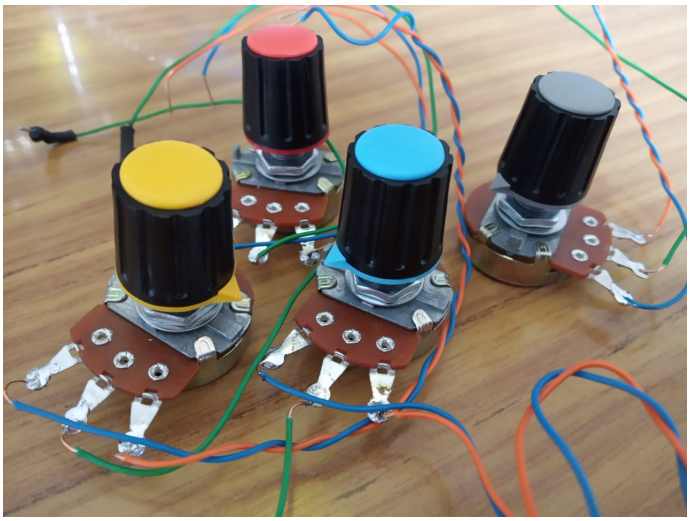
Mozzi es una librería diseñada para usar en microcontroladores Arduino que facilita y permite la generación de sonido mediante un Arduino. Ésta librería brinda algunas facilidades como filtros ya diseñados, distintos osciladores (forma de onda triangular, cuadrada, etc). *Nota*: Ésta librería nos permitió abstraernos un poco mas del problema de cómo hacer sonar al Arduino y nos permitió enfocarnos mas en cómo ejecutar el proyecto, el diseño y funcionalidad esperada.

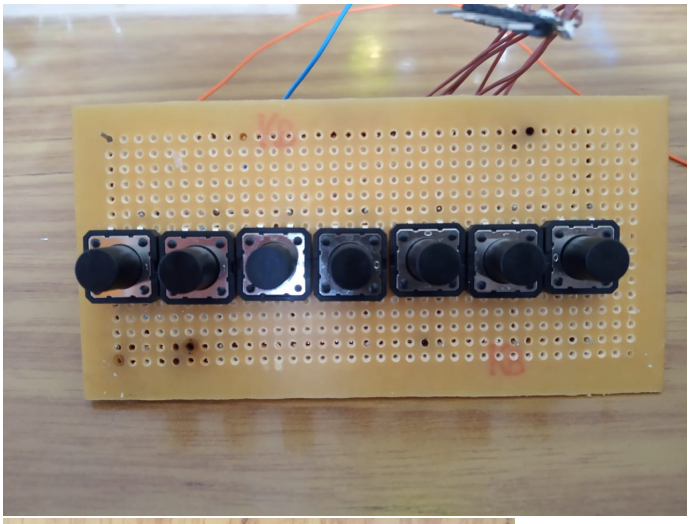
3 Componentes

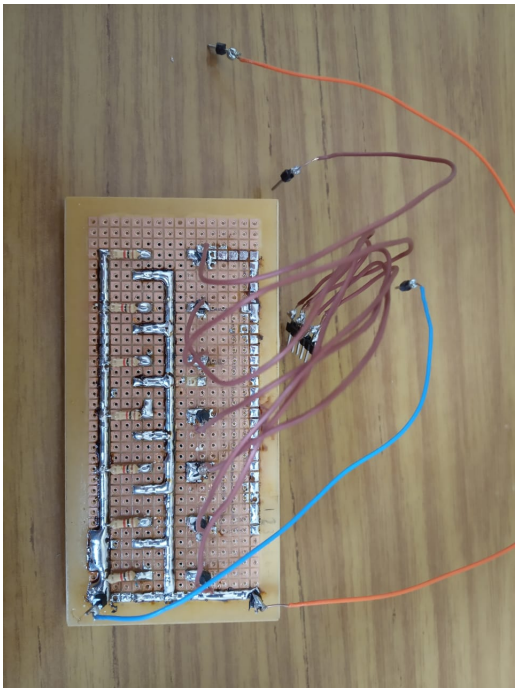
Componentes necesarios para este proyecto:

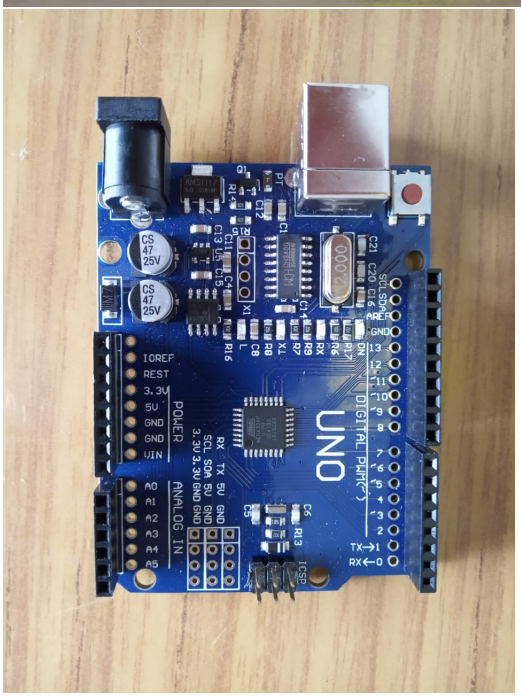
- Arduino UNO

- Protoboard
- Cable USB A-B
- Cable de alambre (los de adentro del cable de red)
- Potenciometro 10 KOhm (4)
- Potenciometro 1 KOhm (1)
- Boton push (7)
- Resistencia 1 KOhm (7)
- Switch palanca (1)
- Jack hembra 3.5 mm
- PCB de prueba









3.1 Software utilizado

El software a utilizar puede ser decidido a gusto de cada uno. En lo personal decidimos utilizar la IDE que brinda Arduino. Sin embargo podriamos utilizar Visual Studio Code, por ejemplo, que cuenta con extensiones para poder compilar y cargar el codigo a la placa. Lo mismo sucede con el sistema operativo. Pasando a detalle:

- Sistema Operativo: Linux (Deepin basado en Ubuntu)
- IDE: Arduino IDE

- Libreria Mozzi

4 Ejecución del proyecto

4.1 Esquemático

4.2 Primeros pasos

Como primer paso, nuestro objetivo era entender el código de otros proyectos que utilizaran la librería Mozzi para poder crear nuestro propio sintetizador de cero, sin copiar código de otros proyectos y saber qué hace cada parte del mismo. A sus vez, leímos parte de la documentación de Mozzi, enfocándonos en las funcionalidades que nos interesaba utilizar, como los filtros y los osciladores principalmente. Esto conllevó tiempo de hacer distintas pruebas con el fin de entender el código y su funcionamiento.

4.3 Comenzando el proyecto

Con una idea mas clara del uso de la librería Mozzi, lo primero que buscamos fue hacer que funcione un oscilador y que la placa Arduino de audio. Necesitamos hacer las conexiones necesarias para disponer de una salida de audio jack 3.5. Con la placa Arduino ya entregando audio, pasamos a agregar un potenciómetro para manipular la frecuencia del oscilador en un rango de frecuencias. Para esto fue necesario conectar el potenciómetro a una de las entradas analógicas de la placa Arduino y mapear los valores entregados al rango de frecuencias que nos interesaba manipular(En este caso la frecuencia de 20Mhz. a 2000 Mhz.).

Repetimos el mismo proceso para agregar un filtro pasa bajos, en ingles, Low pass filter, que filtra las frecuencias altas del sonido, dejando pasar solo las frecuencias bajas. El objetivo era regular el nivel de corte de frecuencias altas, es decir, cuanto mas alto esté el corte, menos frecuencias altas van a sonar, porque están siendo filtradas.

Luego pasamos a agregar distintos osciladores, para lograr tener mas variedad de sonidos. Volvemos a repetir los pasos anteriores para agregar un segundo potenciómetro que permita variar entre los distintos tipos de onda (las utilizadas en este proyecto son triangular, cuadrada, sinusoidal y sierra).

4.4 Agregando teclas

Con el sintetizador básico funcionando decidimos poner botones para actuar como teclas blancas de un piano, pusimos siete, mapeando cada uno con la frecuencia correspondiente a cada nota.

Luego agregamos un switch palanca para poder seleccionar entre modo **sintetizador** y modo **teclado**.

La diferencia es que el primero (**sintetizador**) emite sonido constantemente y a través de los potenciómetros podemos modificar la frecuencia, la forma de onda, el cutoff y el rate de un lfo.

En cambio con el modo **teclado** el potenciómetro de la frecuencia pasa a ser un selector de octavas (C0 a C8) y el sonido solo es emitido cuando se presiona un boton.

5 Problemas

- Muchos de los problemas surgieron al querer agregar funcionalidades como las modulaciones del filtro pasa bajos, y al buscar en internet, las soluciones eran bastantes complejas de entender, por lo que fuimos tomando algunas ideas de dichas soluciones para ir armando nuestro propio código e incorporarlo en nuestro proyecto.
- En la conexiones para la salida de audio Jack 3.5mm. tuvimos problemas de volumen que resultaron provenir de un cable mal conectado a tierra en el componente Jack 3.5 hembra
- En el selector de onda (waveForm), queríamos sobre escribir la onda, y lo que sucedía era que distorsionaba por reproducir los dos tipos de ondas distintos en paralelo, lo corregimos en el código para que se haga el cambio de onda.
- Tuvimos problemas al agregar un botón , ya que lo estábamos conectando de manera errónea, y los esquemas que buscamos en internet eran confusos. Descubrimos que era necesaria una resistencia para las entradas digitales, porque sospechamos que le estábamos enviando mucha energía a la entrada generando que el Arduino se apagara al presionar dicho botón.

6 Código fuente

El código fuente se puede encontrar en https://github.com/HurrellT/JT-Mozzi

Dicho código fue diseñado para ser entendible y para que cualquiera que quiera replicar el proyecto y expandir su funcionalidad pueda hacerlo.

7 Documentación

Podemos encontrar documentación de la librería Mozzi en https://sensorium.github.io/Mozzi/