

# Práctica 3: proyecto Arduino: Auduino, the Lo-Fi granular synthesiser.

Daniel Fernandez Frade  
Ricardo Martinez Robledo

# ¿Qué es un sintetizador?

Un sintetizador es un instrumento musical que, a través de circuitería, genera señales eléctricas que luego se convierten en sonidos audibles. El sintetizador es capaz de generar sonidos creados y modificados, es decir, puede generar sonidos que nosotros mismos hemos creado (que no se parezcan a ningún instrumento) o bien pueden tener el mismo sonido de un instrumento en particular, o bien el sonido de éste modificado. En nuestro caso, el sintetizador es un módulo de sonido, debido a que no posee teclado.



Módulo de sonido.



El sintetizador Early Minimoog.

El sintetizador se empezó a introducir en la música popular a finales de los años 60/principio de los 70. Artistas como Pink Floyd, Jean-Michel Jarre o Alan Parsons utilizaban estos instrumentos en muchas de sus canciones, creando sonidos únicos y cambiando el mundo de la música.



Jean-Michel Jarr en su estudio con sintetizadores.



Pink Floyd en el estudio.



Alan Parsons grabando en el estudio.

## Motivaciones

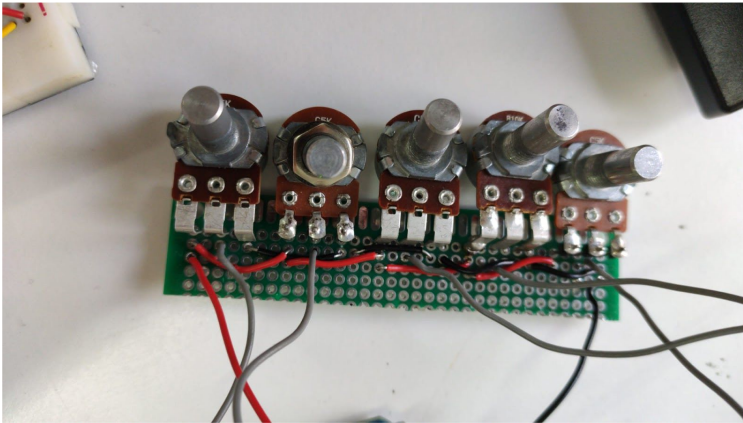
Realmente, hemos decidido realizar este proyecto Arduino debido a nuestra pasión por la música, así como el hecho de que nos parecía un instrumento curioso que reproduce sonidos muy interesantes.

## Estado del arte

Este proyecto de Arduino lo hemos obtenido a partir de un vídeo de youtube ([https://www.youtube.com/watch?v=l1q44jzszwk&ab\\_channel=Nomecopieselnombreplis](https://www.youtube.com/watch?v=l1q44jzszwk&ab_channel=Nomecopieselnombreplis)), donde se nos muestra el montaje y el funcionamiento del mismo. Cabe mencionar que el código de Arduino no pertenece al autor del vídeo, sino que pertenece a Peter Knight, un usuario de Tinker.it que lo proporciona de manera gratuita (se puede encontrar la ayuda del proyecto en el siguiente enlace: <http://code.google.com/p/tinkerit/wiki/Auduino>). El esquema de las conexiones si lo proporciona el autor del vídeo (<http://goo.gl/aW083w>).

# Elementos físicos del Proyecto

Disponemos de 5 potenciómetros tipo b5k los cuales hemos soldado a una placa de prototipado troquelable para poder adaptarlo a nuestras necesidades.

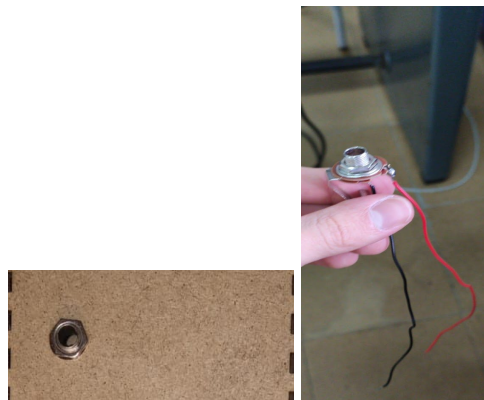


El primero controla el número de veces que la señal se va a repetir, es decir, el número de deltas de la señal.

De los cuatro restantes dos controlan el pitch, uno especializado en frecuencias bajas y otro en frecuencias altas.

Y los otros dos controlan el decaimiento de la señal, el decay uno estaría asociado al pitch 1 y el decay 2 al pitch 2.

La salida del audio la hemos implementado con una salida tipo jack de 6.3 mm con sonido mono.

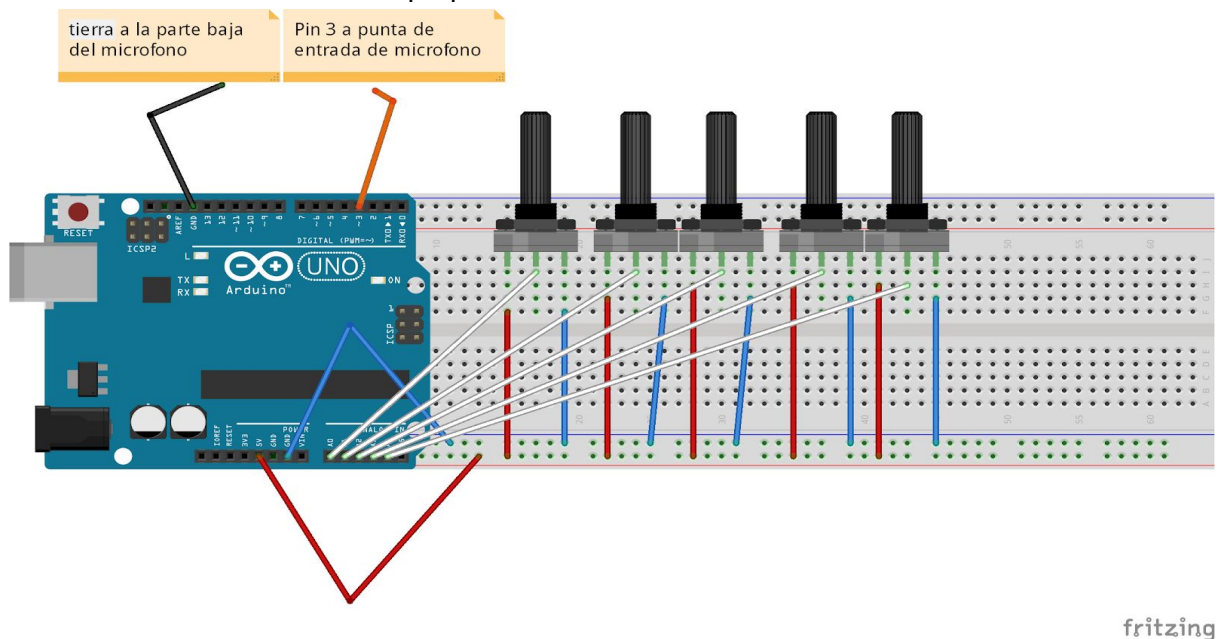


También posee una entrada para la alimentación de 5v y otra para poder conectar Arduino o alimentarlo a través de puerto usb. Gracias a esto podremos modificar el código y cargarlo fácilmente si lo necesitamos.

En cuanto a la caja que hemos diseñado es a base de aglomerado. Para su diseño hemos usado el software box.py y adaptándolo un poco más a nuestras necesidades con el programa Inkscape. Para su fabricación hemos hecho uso de una cortadora láser del fablab SOL. Y gracias a la ayuda de Mario que nos ayudó con el uso de la cortadora láser hemos podido crear la caja.

## Esquemas, gráficos, ficheros o códigos

En primer lugar, para el montaje hemos seguido el diagrama de conexiones que nos proporciona el autor del vídeo:



Podemos observar cómo el circuito se simplifica al uso de 5 potenciómetros que se conectan directamente al Arduino. Las conexiones positivas (cables rojos) van conexas en serie al puerto de 5V, la tierra va conexas en serie al puerto GND, y para el control de los potenciómetros, se conectan a los puertos analógicos (A0-A4). Luego, para la salida, ya sea un jack de audio o directamente un altavoz, se conecta el positivo al puerto 3 del Arduino, y la tierra al puerto GND. En nuestra primera prueba del dispositivo, tuvimos que colocar un amplificador para que el volumen de los sonidos producidos fuera correctamente audible.



En cuanto al código, el autor nos empieza comentando el orden de las conexiones analógicas, siendo analog 0 el perteneciente al pitch 1, el analog 1 al decay 2, el analog 2 al decay 1, el analog 3 al pitch 2 y el analog 4 a la frecuencia de repetición.

Posteriormente, podemos observar que el autor realiza tres mapeados: un mapeado logarítmico, un mapeado midi y un mapeado Pentatónico. En primer lugar, el mapeado logarítmico nos realiza una progresión más lenta para las notas bajas que las altas, para percibir mejor las notas bajas. En segundo lugar, el mapeado midi es el que nos proporciona las notas que va a reproducir el sintetizador, y por último, el mapeado pentatónico es el que nos establece la escala en la que se reproducen estas notas. Esta información no la proporciona el autor, así que, nosotros leyendo e interpretando el código, la hemos deducido.

```
// Smooth logarithmic mapping
//
uint16_t antilogTable[] = {
    64830,64132,63441,62757,62081,61413,60751,60097,59449,58809,58176,57549,56929,56316,55709,55109,
    54515,53928,53347,52773,52204,51642,51085,50535,49991,49452,48920,48393,47871,47356,46846,46341,
    45842,45348,44859,44376,43898,43425,42958,42495,42037,41584,41136,40693,40255,39821,39392,38968,
    38548,38133,37722,37316,36914,36516,36123,35734,35349,34968,34591,34219,33850,33486,33125,32768
};
uint16_t mapPhaseInc(uint16_t input) {
    return (antilogTable[input & 0xf]) >> (input >> 6);
}

// Stepped chromatic mapping
//
uint16_t midiTable[] = {
    17,18,19,20,22,23,24,26,27,29,31,32,34,36,38,41,43,46,48,51,54,58,61,65,69,73,
    77,82,86,92,97,103,109,115,122,129,137,145,154,163,173,183,194,206,218,231,
    244,259,274,291,308,326,346,366,388,411,435,461,489,518,549,581,616,652,691,
    732,776,822,871,923,978,1036,1097,1163,1232,1305,1383,1465,1552,1644,1742,
    1845,1955,2071,2195,2325,2463,2610,2765,2930,3104,3288,3484,3691,3910,4143,
    4389,4650,4927,5220,5530,5859,6207,6577,6968,7382,7821,8286,8779,9301,9854,
    10440,11060,11718,12415,13153,13935,14764,15642,16572,17557,18601,19708,20879,
    22121,23436,24830,26306
};
uint16_t mapMidi(uint16_t input) {
    return (midiTable[(1023-input) >> 3]);
}

// Stepped Pentatonic mapping
//
uint16_t pentatonicTable[54] = {
    0,19,22,26,29,32,38,43,51,58,65,77,86,103,115,129,154,173,206,231,259,308,346,
    411,461,518,616,691,822,923,1036,1232,1383,1644,1845,2071,2463,2765,3288,
    3691,4143,4927,5530,6577,7382,8286,9854,11060,13153,14764,16572,19708,22121,26306
};
```

Luego, el sintetizador funciona mediante un bucle que lo que hace es actualizar los parámetros del oscilador. El autor nos dice que, si empleamos funciones que empleen el uso abusivo de interrupciones, el audio puede ser distorsionado con ruidos no deseados.

El funcionamiento del bucle es simple: en primer lugar, establece el tiempo de inicio de un tren de señales, en segundo lugar, incrementa la fase de estas señales, y finalmente convierte las señales en ondas triangulares. Tras realizar todos estos procesos, se realizarán los mismos para el segundo tren de señales.

```

if (syncPhaseAcc < syncPhaseInc) {
    // Time to start the next grain
    grainPhaseAcc = 0;
    grainAmp = 0x7fff;
    grain2PhaseAcc = 0;
    grain2Amp = 0x7fff;
    LED_PORT ^= 1 << LED_BIT; // Faster than using digitalWrite
}

// Increment the phase of the grain oscillators
grainPhaseAcc += grainPhaseInc;
grain2PhaseAcc += grain2PhaseInc;

// Convert phase into a triangle wave
value = (grainPhaseAcc >> 7) & 0xff;
if (grainPhaseAcc & 0x8000) value = ~value;
// Multiply by current grain amplitude to get sample
output = value * (grainAmp >> 8);

// Repeat for second grain
value = (grain2PhaseAcc >> 7) & 0xff;
if (grain2PhaseAcc & 0x8000) value = ~value;
output += value * (grain2Amp >> 8);

// Make the grain amplitudes decay by a factor every sample (exponential decay)
grainAmp -= (grainAmp >> 8) * grainDecay;
grain2Amp -= (grain2Amp >> 8) * grain2Decay;

// Scale output to the available range, clipping if necessary
output >>= 9;
if (output > 255) output = 255;

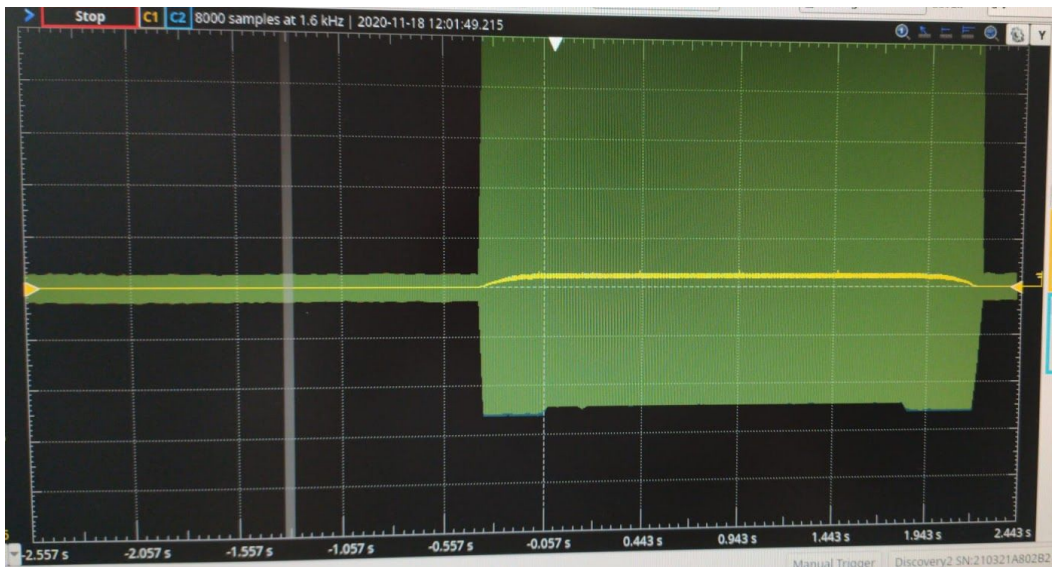
// Output to PWM (this is faster than using analogWrite)
PWM_VALUE = output;
}

```

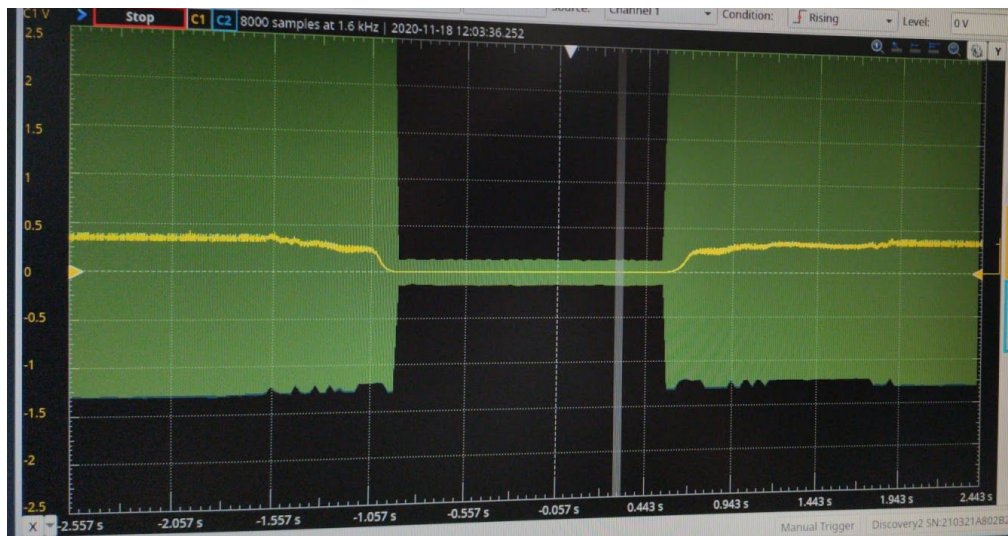
## Resultados obtenidos:

Los resultados obtenidos los hemos analizado con un osciloscopio. Son los siguientes:

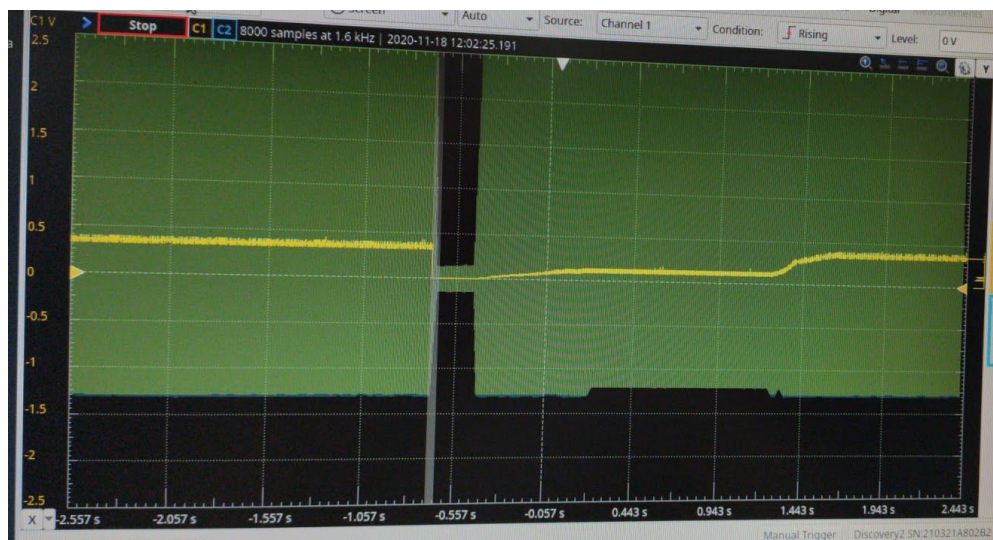
aquí podemos observar como cambia la frecuencia con el pitch 1



y así se ve el cambio en el pitch 2

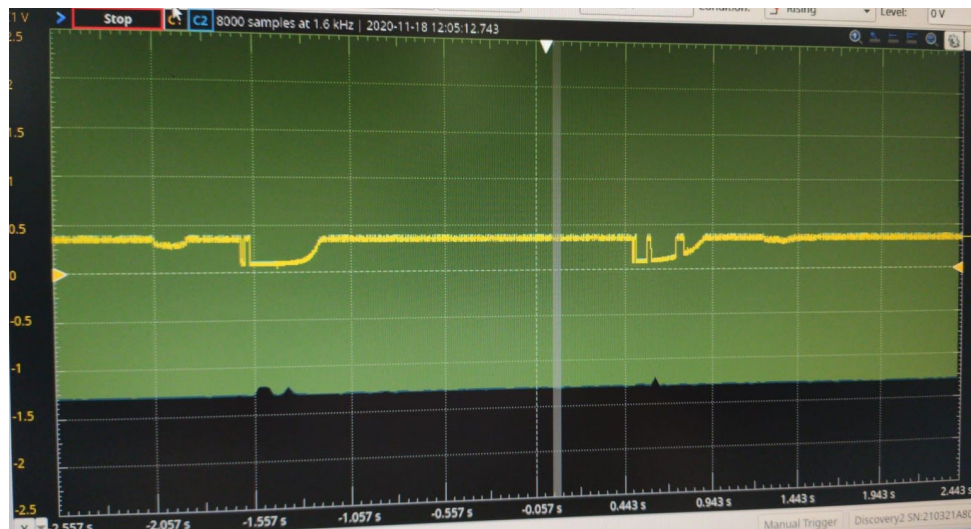


en cuanto a los decays se verían así:  
Decay 1

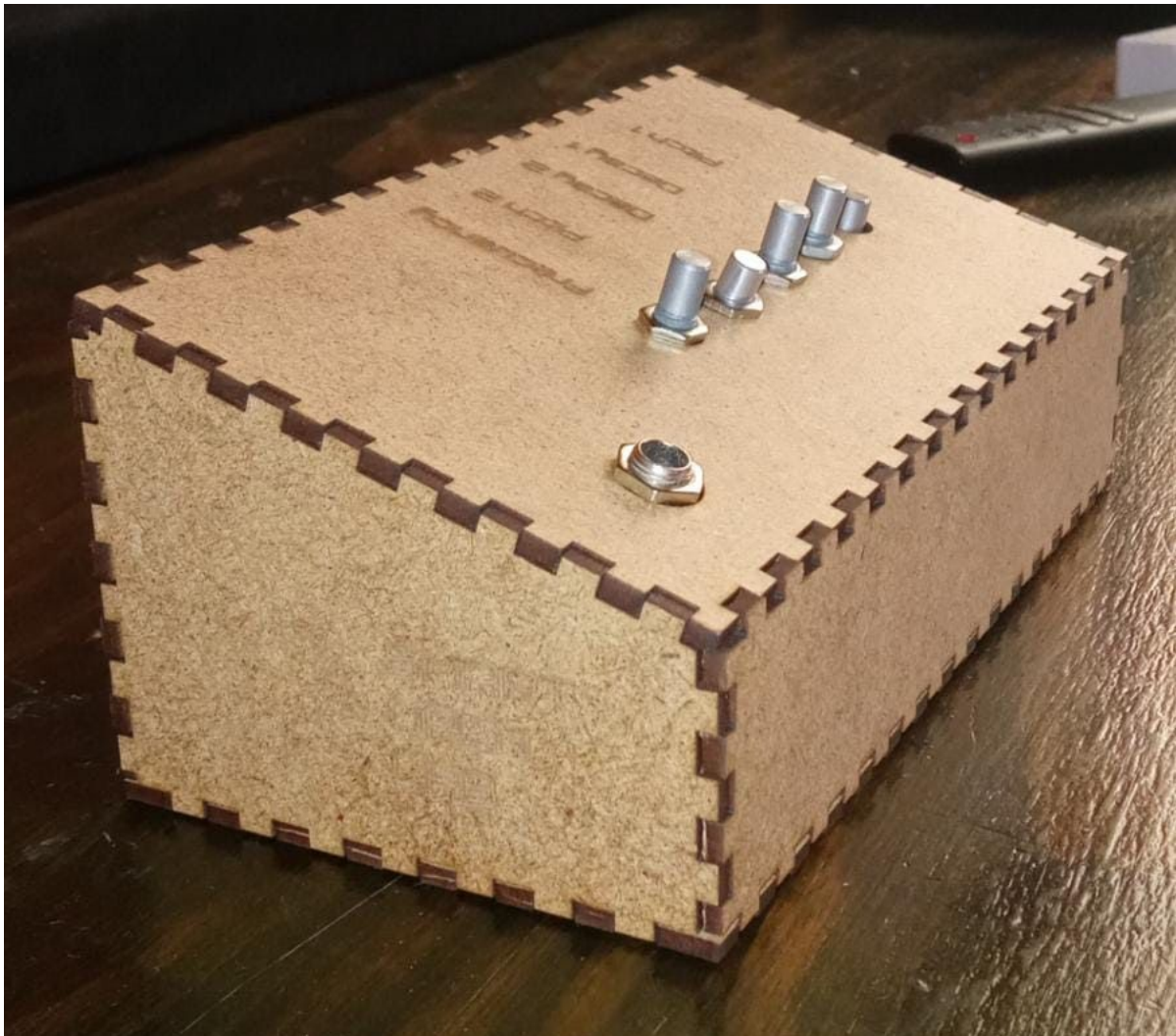


Decay 2





Tras realizar el montaje del sintetizador en su caja, el resultado es el siguiente:



## Futuros desarrollos posibles

En cuanto a los posibles desarrollos futuros se basan desde el cambio del código arduino sobre la función de los potenciómetros hasta la implementación de pads con ritmos que vayan al mismo tempo que las ondas para marcar el beat. También se pueden poner más potenciómetros sin la necesidad de cambiar la caja. O incluso implementar un pequeño teclado de piano. Además, Mario nos dio la idea de incluir una sirena que se reproduzca a la par que las señales que produce el sintetizador. También hemos pensado que es buena idea, para hacer el dispositivo lo más portable posible, incluir un altavoz en la caja. Debido a la falta de volumen del sintetizador conectado directamente a un altavoz, deberíamos incluir un amplificador, tal y como hicimos en nuestras primeras pruebas, pero la idea inicial era incluir un jack para conectarlo a un amplificador directamente. Por último, habría sido interesante realizar este proyecto de tal manera que sea un dispositivo que se conecte entre un instrumento y un amplificador, para producir sonidos únicos.

# Conclusión:

Hemos aprendido mucho en esta práctica. Hemos aprendido a realizar el prototipo de un dispositivo, a interpretar un código de un lenguaje que no conocemos mucho, así como a realizar soldaduras, a utilizar la maquinaria del SmartLab (la cortadora laser, gracias a Mario) y a idear futuras actualizaciones a un proyecto. Además, nos motiva a crear nuevos proyectos, ya sea en Arduino, o en cualquier otro lenguaje, o dispositivos que se basen solo en circuitería.