I.E.S. POLITÉCNICO JESÚS MARÍN

DEPARTAMENTO D E ELECTRÓNICA

PROYECTO FINAL DE MÓDULO

BRAILLE

FRANCISCO MORENO NARANJO

S21ME

Agradecimientos

A mi familia , a mi pareja,

amigos, compañeros,

y profesores.

RESUMEN **3**

JUSTIFICACIÓN DEL PROYECTO: **3**

FINALIDAD DEL PROYECTO: **3**

REVISIÓN BIBLIOGRÁFICA, FUNDAMENTACIÓN TEÓRICA **4**

Historia del braille 4

Objetos similares 5

USOS COTIDIANOS DEL BRAILLE **6**

MATERIALES Y MÉTODOS: **7**

Arduino NANO 7

DFPlayer Mini 7

Convertidor DC-DC HW-106 8

Altavoz de 8 Ohmios 8

Interruptor on/off 8

Resistencias 10k 9

Botones 9

Batería de litio 1200mAh 9

Conector Jack Hembra 10

Placa perforada 10

Cargador de batería TP4056 10

DESARROLLO DEL PROYECTO **11**

ESQUEMAS **21**

DIAGRAMAS DE FLUJO **22**

CÓDIGO DEL PROYECTO **23**

CONCLUSIONES Y PROPUESTAS DE MEJORA **29**

WEBGRAFÍA **30**

RESUMEN

A continuación se expondrán todos los datos sobre mi proyecto final de módulo, Braille, el cual consiste en la realización de un juguete para niños con discapacidad visual para que a la hora de aprender a usar el lenguaje braille les sea más sencillo y ameno. Pero ¿Qué es el Braille?
El braille es un sistema de lectura y escritura táctil pensado para personas ciegas. Se conoce también como cecografía. Fue ideado a mediados del siglo XIX por el francés Louis Braille. Cuando tenía 13 años, el director de la Escuela de Ciegos y Sordos de París le pidió que probara un sistema de lectoescritura táctil inventado por un militar para transmitir órdenes a puestos de avanzada sin tener necesidad de delatar la posición durante las noches. Louis Braille descubrió al cabo de un tiempo que el sistema era válido y lo reinventó utilizando un sistema de 8 puntos. Al cabo de unos años lo simplificó dejándole en el sistema universalmente conocido y adoptado de 6 puntos.
Mi trabajo se basa en la implementación de la electrónica para la facilitación del aprendizaje de estos niños con un juguete creado desde 0 en el que se introducirán los 6 botones para la formación del alfabeto y 2 botones extras para un modo de juego random y en solitario del alumno y otro para la comprobación de las letras introducidas.

JUSTIFICACIÓN DEL PROYECTO:

Cuando vinieron los maestros de Pedagogía Terapéutica, que trabajan de apoyo a alumnado con discapacidad y expusieron el trabajo se me imaginaron mil ideas que hacer y cómo enfocarlo de diferentes maneras, fue un proyecto que me llamó la atención y pensé en pedirlo voluntariamente ya que tenía muchas ganas de realizarlo, mi proyecto del módulo de empresa también intenta ayudar a gente con discapacidades auditivas y niños por lo que pensé que este trabajo encajaría muy bien conmigo y cuando se me propuso hacerlo no dudé en aceptarlo para ayudar a los niños y que tuvieran otro método de aprendizaje más adaptado a sus edades y a la tecnología que hay hoy día.

FINALIDAD DEL PROYECTO:

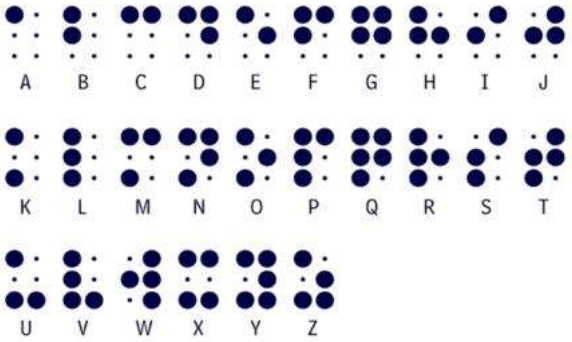
La finalidad del proyecto es la de facilitarle las cosas a los niños con discapacidad visual y hacerles más ameno la enseñanza a los profesores que tienen ya que el no disponer de objetos para enseñanza y los problemas que acarrean estos es normal que el niño/a pierda interés, sin embargo, en este proyecto se prima el aprendizaje y la solución a los problemas de estos niños siendo ellos también participe de ello, ya que los audios que se ejecutan con el proyecto son grabados y realizados por ellos por lo que les hará sentir bien y será una forma de contribuir y formar parte de esto.

REVISIÓN BIBLIOGRÁFICA, FUNDAMENTACIÓN TEÓRICA

Historia del braille

Es el principal medio de comunicación de una persona con discapacidad visual profunda, por lo tanto haremos hincapié en estas primeras páginas a este sistema. Es información sobre y para personas ciegas o con discapacidad visual.

El sistema braille, inventado en el siglo XIX, está basado en un símbolo formado por 6 puntos: aquellos que estén en relieve representan una letra o signo de la escritura en caracteres visuales. El tamaño y distribución de los 6 puntos forman el llamado Signo Generador. Las terminaciones nerviosas de la yema del dedo captan este tamaño. Este signo sólo permite 64 combinaciones de puntos.

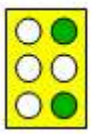


- Los puntos negros pequeños son los puntos del signo generador que no están en relieve, sólo se dibujaron para una mejor comprensión de cada símbolo.
- Los símbolos correspondientes a la primera fila ocupan sólo los cuatro puntos superiores del signo generador.
- En la segunda fila son iguales a los de la primera, pero se le agrega el punto inferior izquierdo (salvo en la "ñ" que es propia del idioma español).
- En los de la tercera se agregan los dos inferiores.

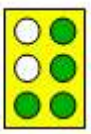
Símbolos Dobles

Los llamados símbolos dobles dan un significado a la letra que anteceden y es este el símbolo que se utiliza, lo podemos ver aquí abajo cómo es y un ejemplo de cómo se utiliza:

Signo de mayúscula. Para representar las letras mayúsculas se antepone el carácter braille formado por los puntos 4 y 6.
Signo de número. Para representar los números se antepone el carácter braille formado por los puntos 3,4,5 y 6.



Signo de mayúscula



Signo de número

Objetos similares

En este apartado se hará una revisión del tema sobre cosas ya publicadas o realizadas en el ámbito del trabajo.

Braillin

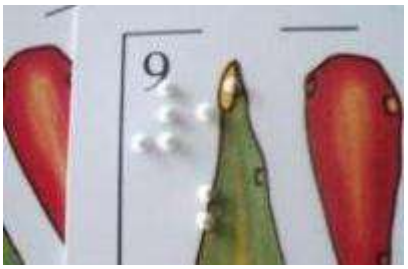
Este juguete es parte de la idea del proyecto el cual ha sido el generador de la idea ya que se buscaba algo similar pero que fuera interactivo no un muñeco sin más con el que se podría interactuar pero no da respuesta, el usuario no sabría si lo está haciendo bien o mal ya que no consigue respuesta y tendrían que estar los profesores encima para ir comprobando uno por uno si lo están haciendo bien con mi proyecto eso no sería necesario ya que el propio proyecto te diría si lo estas haciendo bien o mal por lo que se le ahorra trabajo a los profesores y hace a los usuarios más independientes a la hora de aprender.

Este es el otro modo de enseñanza que tienen los niños sin embargo es algo muy ambiguo y muy simple lo cual en mi opinión no transmite nada al usuario y hace la enseñanza monótona

Este podría ser el juguete más parecido a mi proyecto ya que es interactivo y cuenta con sistemas de electrónica, se puede usar tanto para leer pequeñas palabras como para aprender las letras y es de bajo coste. Está realizado por un ingeniero informático con una hija con discapacidad visual.

USOS COTIDIANOS DEL BRAILLE

El braille es un lenguaje que está presente hoy día en nuestra sociedad sin embargo suele pasar desapercibido para las personas videntes ya que no se tiene en cuenta, por ello se expondrán algunos casos en los que el braille está presente.



Cartas y juegos de mesas con anotaciones en braille para la facilitación de estos juegos a personas con discapacidad visual. Algo que hace que las personas con discapacidad visual se puedan integrar en los juegos de mesa.

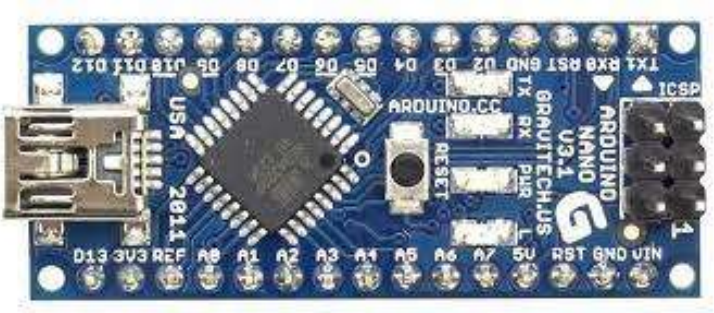
Uno de los usos más presentes del braille y donde más destaca es en los ascensores, donde se puede diferenciar claramente el uso de este lenguaje, se puede observar como tiene el prefijo de número y el número correspondiente para la selección de planta y los prefijos necesarios para las diferentes opciones que tienen los ascensores.

El uso del braille en las cajas de medicamentos es algo que lleva usándose desde hace mucho tiempo ya que personas mayores sufren de pérdida de visión y estos son los que más uso le dan y tiene que estar adaptado para todas las personas

En algunos países la impresión de billetes con marcas para invidentes se está implementando para facilitarle a las personas con discapacidad visual el uso del dinero.

MATERIALES Y MÉTODOS:

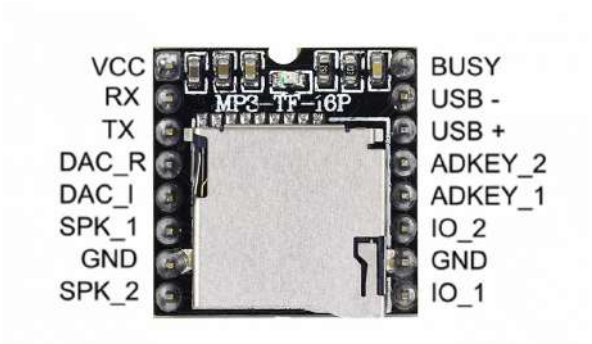
Arduino NANO



La Nano basada en la ATmega328 (Arduino Nano 3.x) incorpora la ATmega328P, la misma que el Arduino UNO. La principal diferencia entre ellos es que el Arduino UNO se presenta en forma de PDIP (Plastic Dual-In-line Package) con 30 pines y la Arduino Nano está disponible en TQFP (plastic quad flat pack) con 32 pines. Los 2 pines adicionales de la Arduino Nano sirven para las funcionalidades del ADC, mientras que UNO tiene 6 puertos ADC pero la Nano tiene 8 puertos ADC.

La placa Arduino Nano no tiene una toma de alimentación DC como otras placas Arduino, sino que tiene un puerto mini-USB. Este puerto se utiliza tanto para la programación como para la monitorización en serie. La característica fascinante de Nano es que elegirá la fuente de energía más fuerte con su diferencia de potencial. En nuestro proyecto el arduino NANO nos servirá como memoria y controlador a él irán conectado todos nuestros cables ya que será el encargado de organizar y llevar a cabo todo el funcionamiento del proyecto

DFPlayer Mini



El DFPlayer Mini es un reproductor de audio de bajo coste y pequeño que podemos conectar a un procesador como Arduino para reproducir audio en formato MP3. Será el controlador de audio el cual llevará incluida una tarjeta SD en la cual irán todos los audios que se reproduzcan por el altavoz. Los audios tienen que llevar una posición y un nombre concreto para que funcione correctamente. En nuestro proyecto se usarán los pines de VCC RX TX SPK_1 GND y SPK_2. El DFPlayer Mini se conecta a traves de los pines de RX y TX que son de donde se capta la información del Arduino NANO los cuales están conectados al puerto serie que es su principal comunicación.

Convertidor DC-DC HW-106

Es un tipo de convertidor de potencia que transforma corriente continua de un nivel de tensión a otro, en nuestro caso se usará para convertir los 3,7V que salen de la batería a los 5V que necesita el arduino NANO para alimentarse.

Altavoz de 8 Ohmios

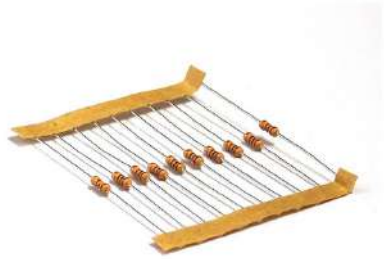
A través de los dos pines de conexión que posee el DFPlayer Mini lo que produce una conversión de energía eléctrica en energía mecánica produciendo así los sonidos correspondientes a través del campo magnético producido por la bobina.
Los altavoces funcionan al convertir energía eléctrica en energía mecánica.
La energía mecánica comprime el aire y convierte el movimiento en energía sonora o nivel de presión sonora (SPL).

Altavoz de 8 Ohms y 0.5W

Interruptor on/off

Interruptor de dos modos on/off que será usado para encender y apagar el proyecto. Su conexión será a través del convertidor de DC-DC

Resistencias 10k



Usadas para ajustar los valores en los circuitos y el DFPlayer Mini

Botones

Botones se usarán 6 de anclaje para crear los códigos en el sistema braille de color rojo y 2 pulsadores amarillos para comprobar los resultados. Donde el izquierdo funcionará para comprobar la letra introducida y el derecho será un modo de juego de letras random que hasta que no se acierte no salga de este si no es encendiendo y apagando el dispositivo

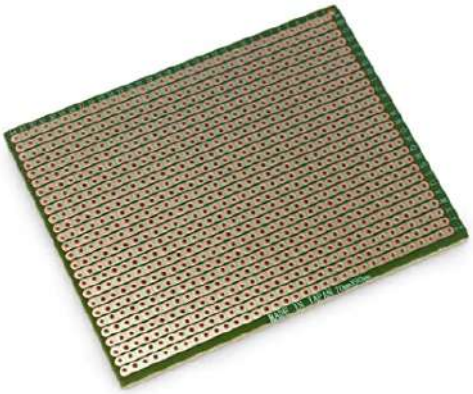
Batería de litio 1200mAh

La batería de iones de litio, es un dispositivo con dos o tres celdas de energía diseñado para el almacenamiento de energía eléctrica que emplea como electrolito una sal de litio que consigue los iones necesarios para la reacción electroquímica reversible que tiene lugar entre el cátodo y el ánodo.
Será la encargada de suministrar energía al proyecto y tiene una sola celda de 3.7V.

Conector Jack Hembra

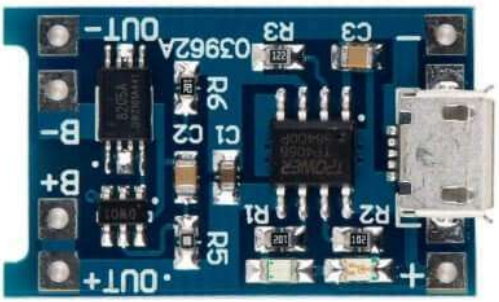
Se usará para la implementación de un conector jack al proyecto el cual podrá ser usado para el uso de auriculares para evitar el ruido y poder centrarse mejor.

Placa perforada



Placa la cual consta de agujeros y pines conectados entre sí por vías de cobre para la facilitación de la conexión de los diferentes circuitos y conexiones del proyecto

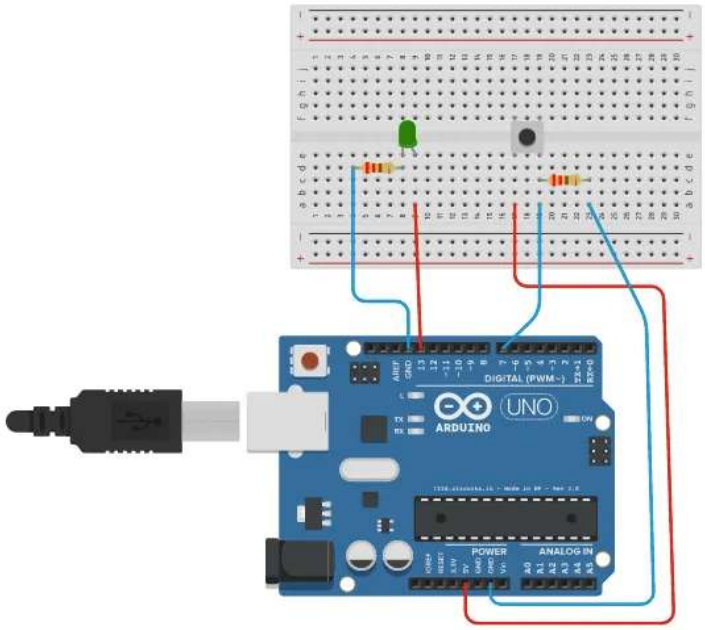
Cargador de batería TP4056



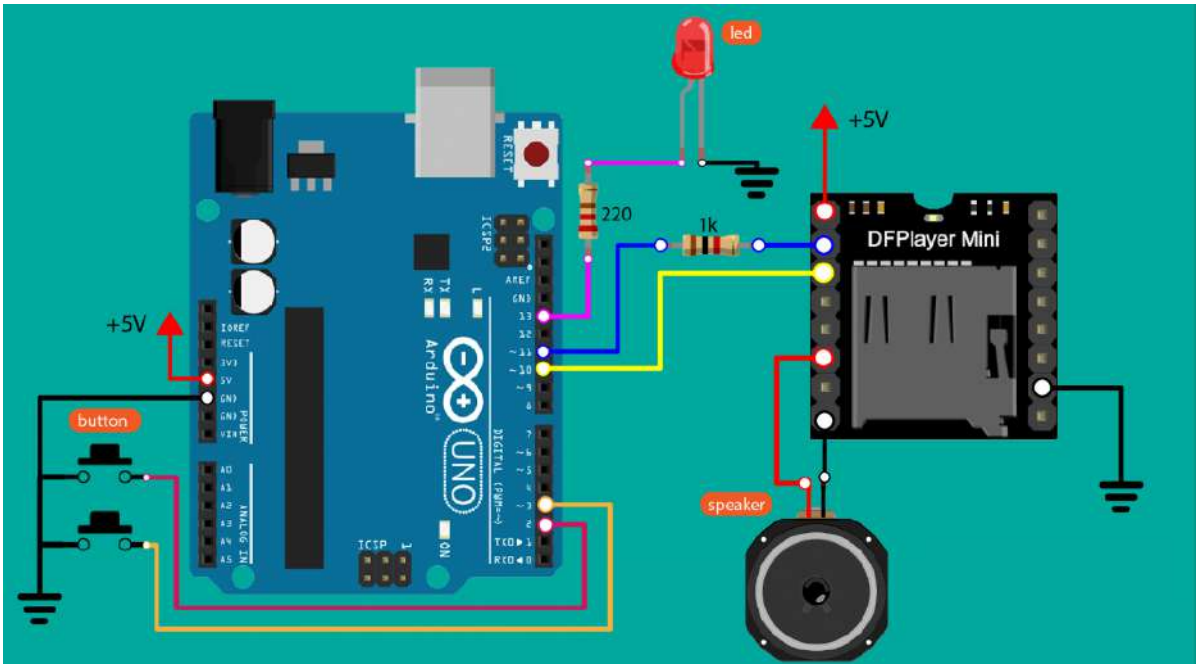
Este módulo ofrecerá una corriente de carga de 1A y luego se cortará cuando haya terminado la carga, lo que ayudará a optimizar el tiempo de vida útil de tu batería. Además, cuando el voltaje de la batería cae por debajo de 2,4 V, el chip de protección desconectará la carga para proteger la celda de funcionar a una tensión demasiado baja y también protege contra la conexión de sobretensión y polaridad inversa (normalmente se destruirá en lugar de batería).
Voltaje de entrada: 4V-8V, salida de corriente de carga máxima: 1000mA

DESARROLLO DEL PROYECTO

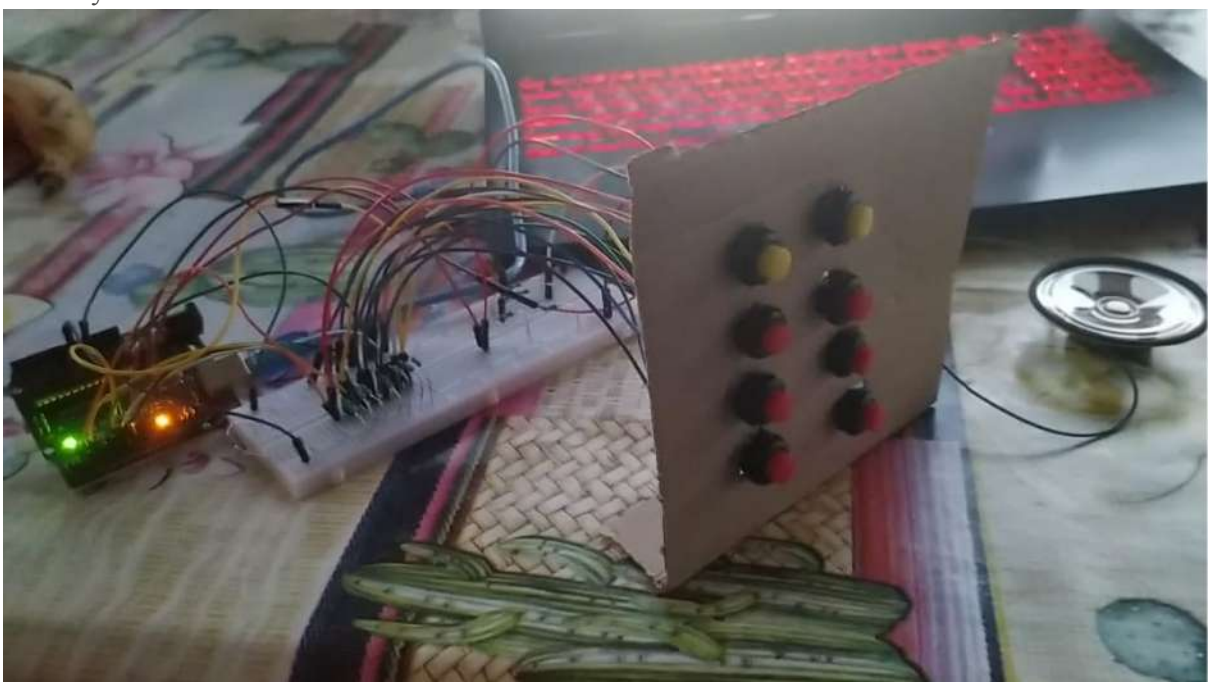
A continuación se explicarán los pasos seguidos y los diagramas de conexionado del proyecto. Primero busqué información en internet sobre el funcionamiento de los botones y los pulsadores e hice programas sencillos para comprender su funcionamiento, encendiendo y apagando un LED.

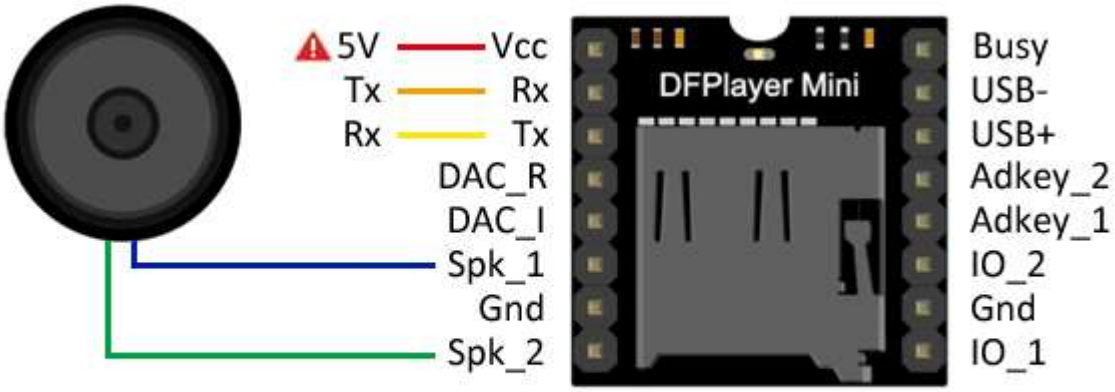


Una vez entendido esto busqué información sobre el DFPlayer Mini e hice programas sencillos hasta que conseguí entenderlo y reproducir los sonidos que yo quisiera. A continuación junté las dos partes e hice un programa donde se pasaba de canción cuando se pulsa un botón, todo esto lo realice como inicio del proyecto para entender el funcionamiento de las diferentes partes y con un arduino convencional.



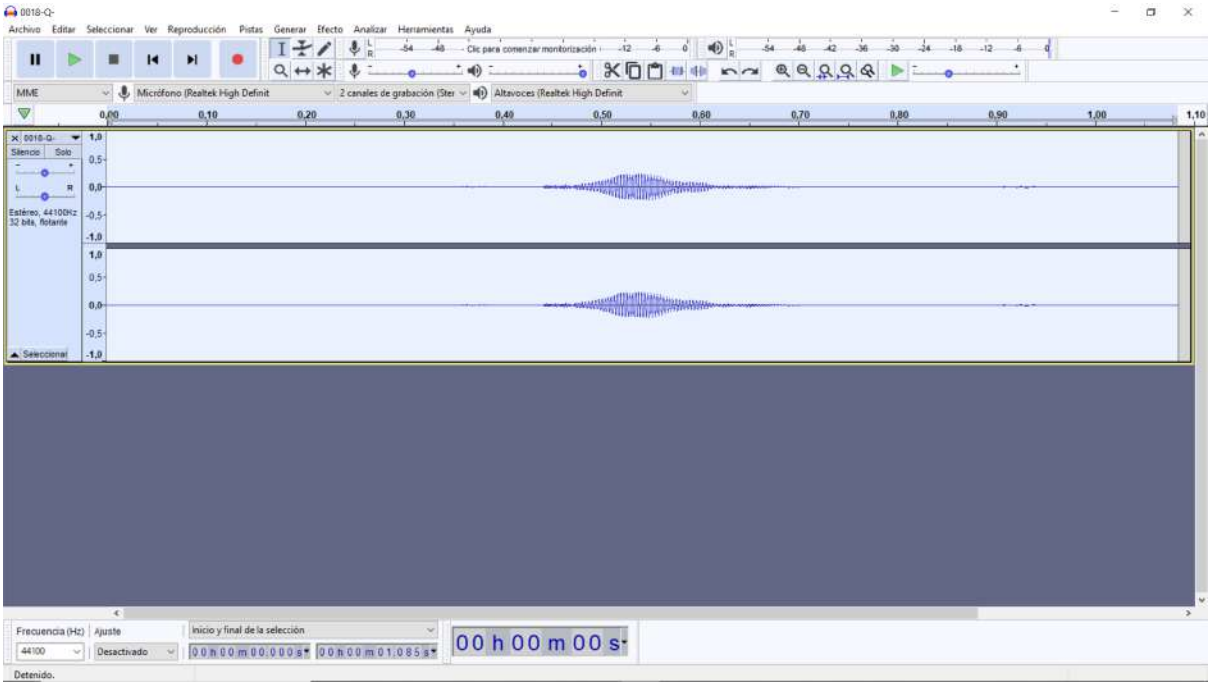
Una vez comprendido esto monté un prototipo donde estaban los botones, el DFPlayer Mini, el altavoz y el arduino uno.



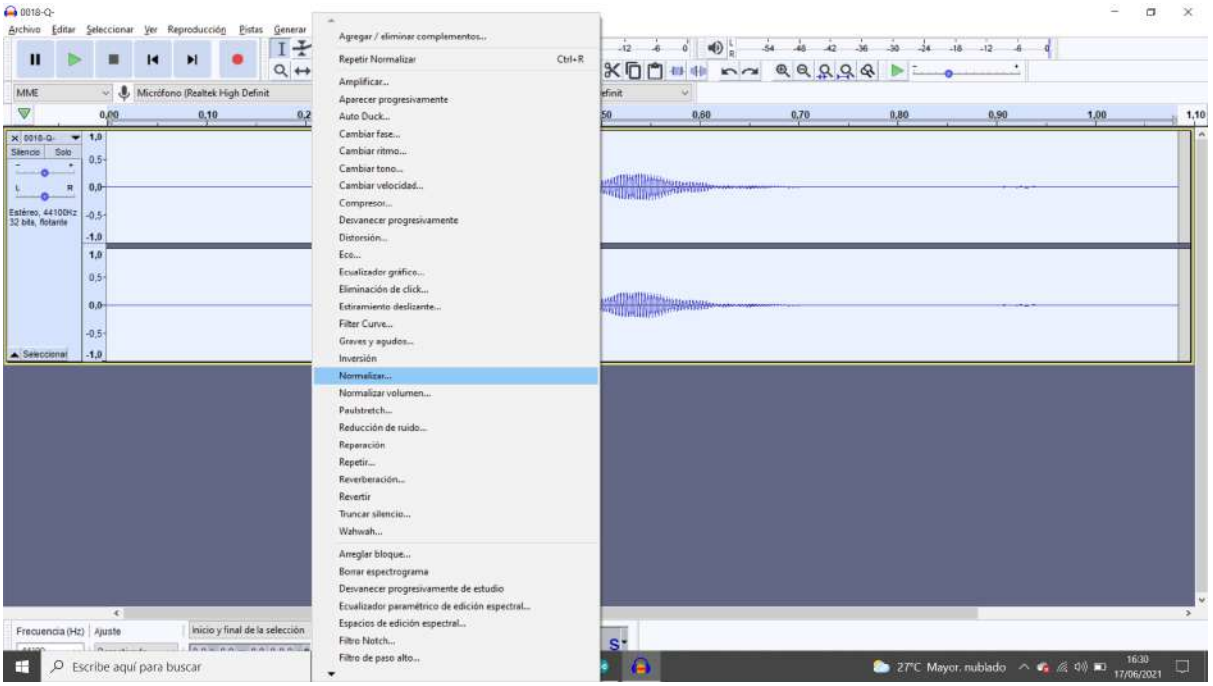


Esta es la conexión usada para conectar el DFPlayer Mini al arduino y al altavoz. Este pequeño módulo es capaz de reproducir audio en formato .mp3 y .wav. Sin embargo, al nombre de la canción previamente se tiene que agregar un número de 4 dígitos; ejemplo: “0001audio1.mp3”. Por otro lado, este dispositivo puede leer carpetas, no obstante, también al nombre de la carpeta previamente se tiene que agregar un número de 2 dígitos; ejemplo: “01 instrumentales”. Finalmente, tenemos que tener en cuenta que este reproductor soporta hasta 100 carpetas y cada carpeta tiene que tener un máximo de 225 canciones o audios. He encontrado fallos a la hora de introducir los audios en la tarjeta SD y nombrarlos por lo que hay que introducir los audios en el orden en el que se ejecutarán ya que si no la memoria del formato no lo tiene en cuenta y no cumple con lo especificado, esto es una aclaración por si en el futuro se quieren cambiar los audios.

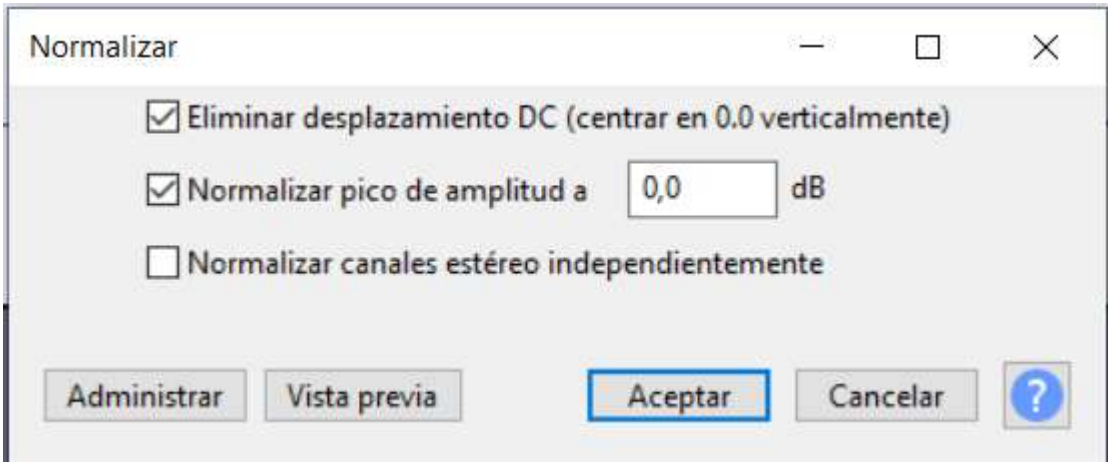
A continuación recibí los audios que deberían de ponerse en el proyecto y al probarlos noté que estaban muy bajos algunos y otros altos por los que hice una normalización en todos los sonidos del proyecto para poder tenerlos todos al mismo volumen, el programa utilizado ha sido Audacity.

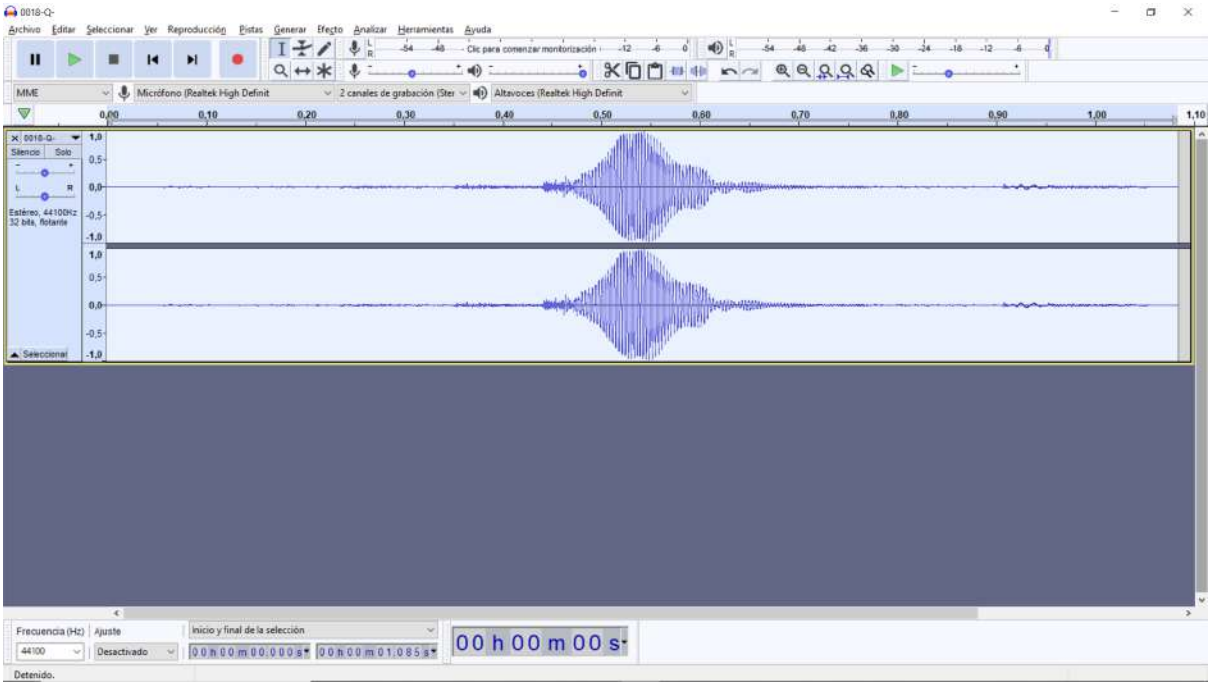


Audio de la letra Q sin normalizar, se selecciona el audio con la pestaña de seleccionar (esquina inferior izquierda) o con las teclas control + A.

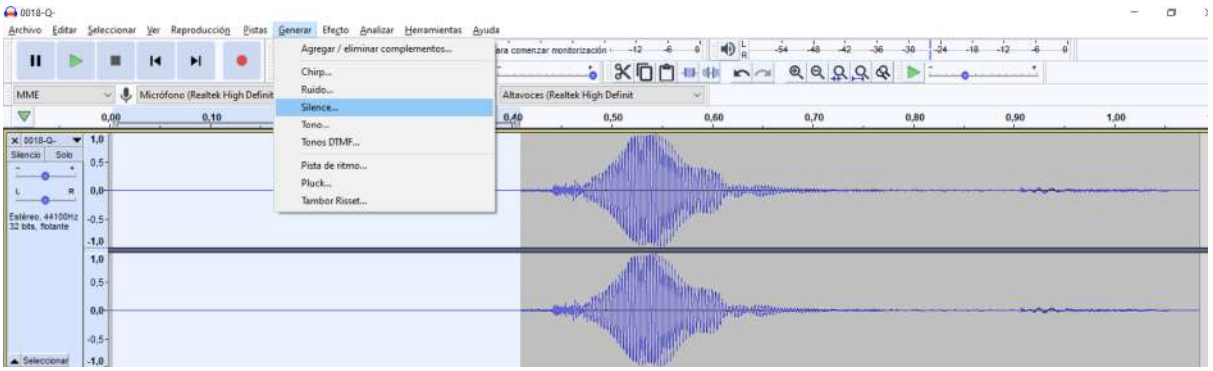


A continuación se le da al apartado efecto→ normalizar y se le indican 0 dBs





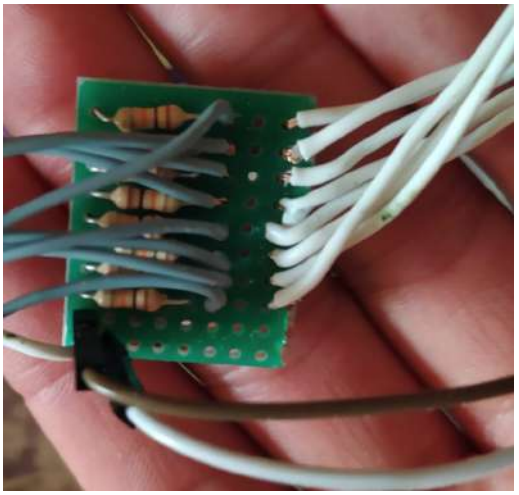
Una vez hecho esto le damos a aceptar y se nos genera la normalización del audio, al hacer esto la mayoría de sonidos tenían ruido tanto al principio como al final de los audios por lo que había que quitarlos y el propio programa nos facilita esta herramienta, para ello habrá que seleccionar la parte donde está el ruido y darle a generar y luego a silence lo cual nos quitará el ruido de la zona seleccionada.



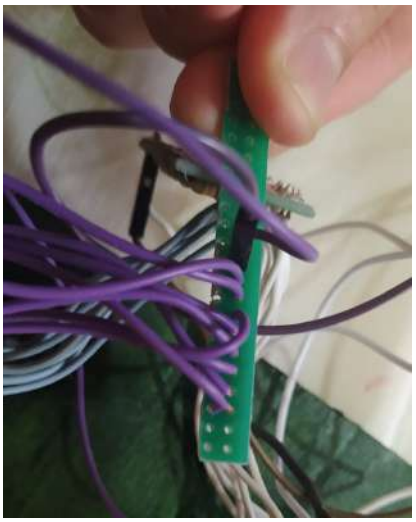
Una vez todos los audios normalizados podemos poner el volumen en el programa general ya que todos estarán a la misma intensidad.

Con todo el proyecto montado en el prototipo y en funcionamiento me dispongo a trasladar todo lo hecho a su forma final.

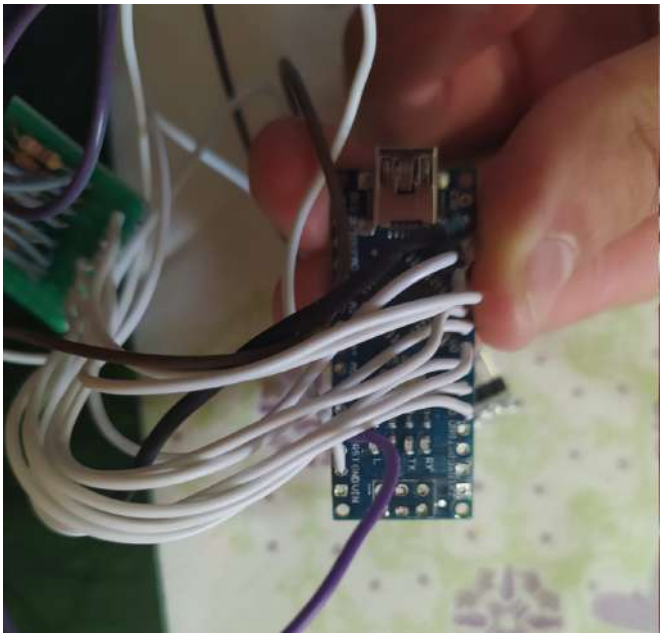
Lo primero que hacemos es buscar una madera donde poder colocar los botones, se hacen los agujeros con un taladro y se introducen los botones correspondientes y se les sueldan sus pines a tierra y a 5V con sus respectivas resistencias, para ello usamos una placa perforada para que las soldaduras estén más protegidas y no estén al aire.



También usamos una placa perforada para hacer todas las conexiones a 5V ya que no es solo una y para el mejor conexionado es lo más óptimo.



Por otro lado, todos los botones, la masa común y todas las conexiones se harán en la placa de Arduino NANO.



A continuación preparamos el chasis e instalamos escuadras para fortalecer el agarre. También pintamos la madera de verde para darle tonalidad y no sea un diseño muy apagado.

Sin embargo encontramos un fallo a la hora de hacer esto, los tornillos atraviesan la madera por lo que podría ser peligroso para los usuarios por lo que optamos por hacerlos desde fuera y utilizarlos también como forma decorativa.

Aquí podemos observar los tornillos sobresalidos y la alternativa que se ha buscado, para que tenga más fuerza se incluirá silicona caliente por dentro del chasis para reforzarlo. Se pueden observar ambas variantes y como queda por fuera los anclajes buscados.

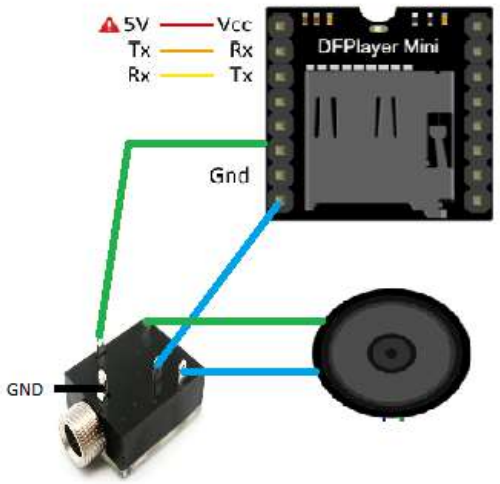
Para mejorar el sonido del proyecto se han realizado agujeros en la parte del altavoz para que el usuario escuche los audios con facilidad y no tenga un sonido muy bajo o cueste escucharlo. Los agujeros se han realizado con un taladro. También se le ha añadido el botón de encendido/apagado al frontal del proyecto para que el usuario tenga todos los mecanismos más asequibles y no dificulte su utilización.

Para el acceso del técnico por si se quieren hacer cambios o cambiar los audios del proyecto se le incluirán bisagras en la parte de arriba para facilitar su acceso, también por temas de espacio se ha recortado un cable usb a mini usb y soldado para que cupiera correctamente y no quitará mucho espacio a la parte de dentro.

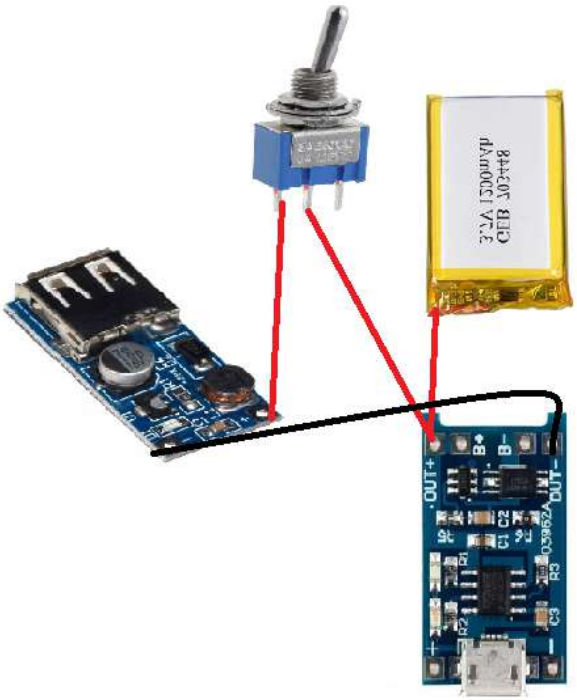
Toda la parte interior ha sido puesta con silicona para evitar su desplazamiento a excepción de una de las placas perforadas para permitir así el poder abrirlo con las bisagras para acceder a la parte interior.

ESQUEMAS

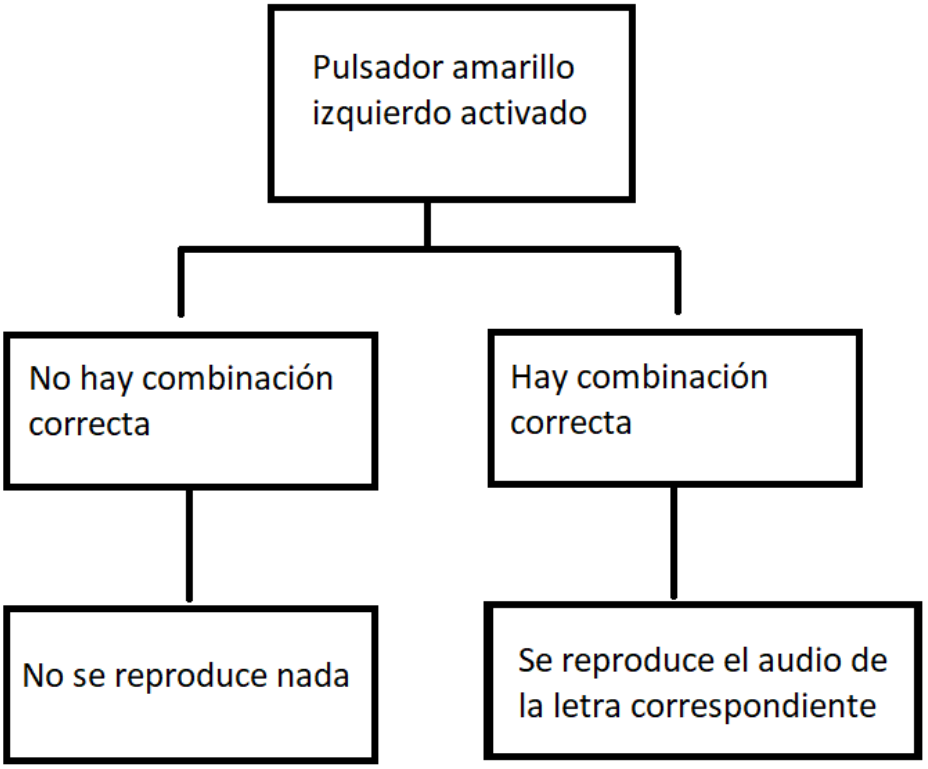
Esquema de conexionado para el conector jack, para comprobar cuál eran los pines de entrada y de salida del conector se han conectado unos auriculares y con estos puestos y un polimetro en continuidad se ha ido dando contacto en los pines hasta que se origina un pequeño sonido el cual indica que ese pin es de entrada.

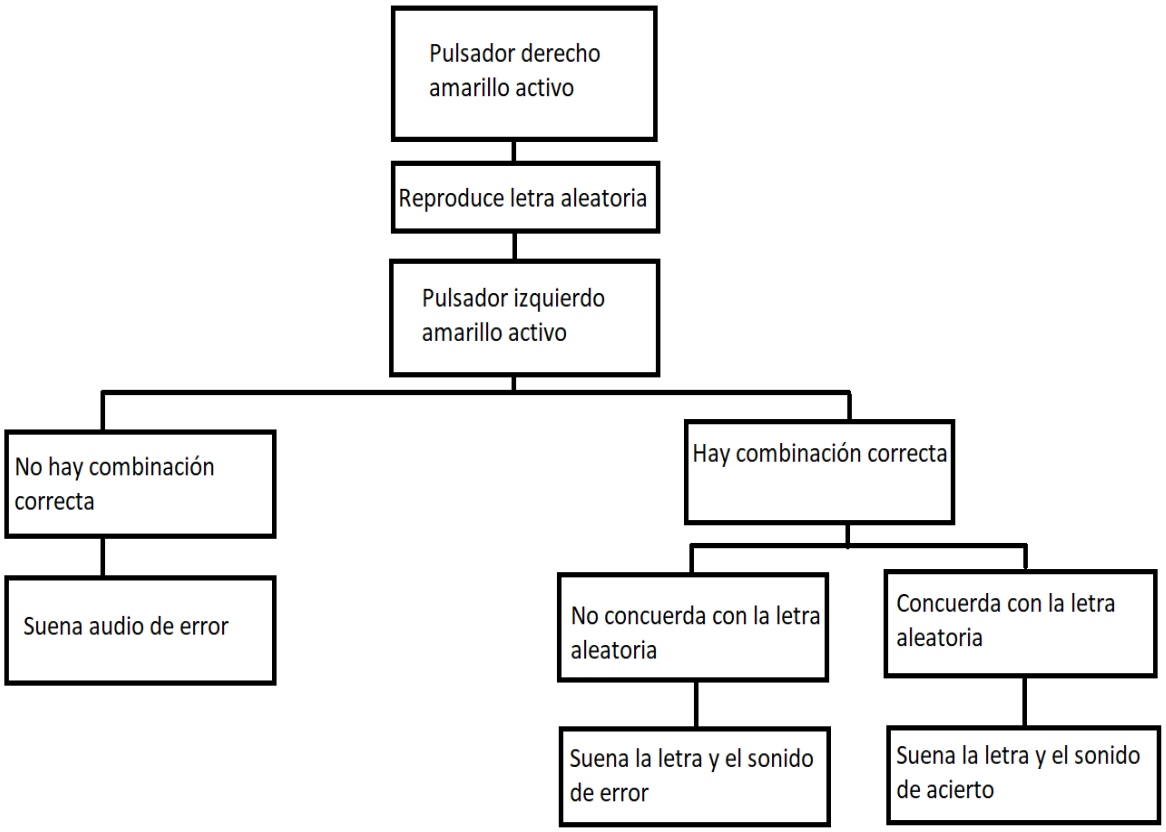


Aquí podemos ver el conexionado de la parte de alimentación la cual está compuesta por un interruptor de palanca on/off, el convertidor DC-DC, el cargador de batería y la batería. Se usa el interruptor para la alimentación al convertidor de DC-DC y se pueda encender y apagar el proyecto.



DIAGRAMAS DE FLUJO





CÓDIGO DEL PROYECTO

```
#include "Arduino.h"
#include "SoftwareSerial.h"                    //inclusión de librerías
#include "DFRobotDFPlayerMini.h"

int Nrandom;
int comparator=0;
int i=0;
int pulsador=0;                  //Declaración de todas las variables que se van a usar
int pulsador1=0;
int pulsador2=0;
int pulsador3=0;
int pulsador4=0;
int pulsador5=0;
int pulsador6=0;
int pulsador7=0;
SoftwareSerial mySoftwareSerial(10, 11);    //  Inicialización del DFPlayer y todas sus
DFRobotDFPlayerMini myDFPlayer;              instrucciones necesarias
void printDetail(uint8_t type, int value);
void setup()
{

  randomSeed(analogRead(A0));        //lee un valor del pin analogico A0 que es aleatorio
  pinMode(2, INPUT);                 //declaramos el pin 2 como entrada
  pinMode(3, INPUT);                 //declaramos el pin 3 como entrada
  pinMode(4, INPUT);                 //declaramos el pin 4 como entrada
  pinMode(5, INPUT);                 //declaramos el pin 5 como entrada
  pinMode(6, INPUT);                 //declaramos el pin 6 como entrada
  pinMode(7, INPUT);                 //declaramos el pin 7 como entrada
  pinMode(8, INPUT);                 //declaramos el pin 8 como entrada
  pinMode(9, INPUT);                 //declaramos el pin 9 como entrada



  mySoftwareSerial.begin(9600);              //inicialización del puerto serie
  Serial.begin(115200);
  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));
  if (!myDFPlayer.begin(mySoftwareSerial)) //Uso del puerto serie para la comunicación con
{                                                  mp3
 Serial.println(F("Unable to begin:"));            //avisos del DFPlayer si hay algo mal conectado
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true);
  }
```

```
  Serial.println(F("DFPlayer Mini online."));
  myDFPlayer.setTimeOut(500);   //Selecciona la comunicacIón con el puerto serie cada 500ms
  myDFPlayer.volume(25);  //Selección del volumen (0~30).
  myDFPlayer.volumeUp(); //Subida de volumen
  myDFPlayer.volumeDown(); //Bajada de volumen
  myDFPlayer.EQ(DFPLAYER_EQ_NORMAL);  //equalizacion
  myDFPlayer.outputDevice(DFPLAYER_DEVICE_SD);   // uso de tarjeta sd

  Serial.println(myDFPlayer.readState()); //Lee el estado del mp3
  Serial.println(myDFPlayer.readVolume()); //Lee el volumen
  Serial.println(myDFPlayer.readEQ()); //Lee la configuración de equalizacion
  Serial.println(myDFPlayer.readFileCounts()); //Lee los archivos de la SD
  Serial.println(myDFPlayer.readCurrentFileNumber()); //Lee el número de archivo de
reproducción actual
}


void loop()
{

  pulsador = digitalRead(9);   //lee el estado de los botón
  pulsador1=digitalRead(8);
  pulsador2=digitalRead(7);
  pulsador3=digitalRead(6);
  pulsador4=digitalRead(5);
  pulsador5=digitalRead(4);
  pulsador6=digitalRead(3);
  pulsador7=digitalRead(2);

  if(pulsador7==HIGH)     //función para el modo juego generando un valor random de 1,27
    {
      i=1;

      Nrandom = random(1,27);
       myDFPlayer.play(Nrandom);
       delay(1000);
       comparator=0;
    }

  if(pulsador==HIGH) //si el estado es pulsado   //Dependiendo de los botones rojos pulsados al
aacionar el pulsador izquierdo busca entre todos los casos el correspondiente y ejecuta el audio
  {
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==LOW && pulsador4==LOW
&& pulsador5==LOW && pulsador6==LOW)
    {
      comparator=1;
     myDFPlayer.play(1);  //A
       delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==LOW && pulsador4==LOW
&& pulsador5==LOW && pulsador6==LOW)
    {
      comparator=2;
      myDFPlayer.play(2);  //B
      delay(1000);
```

```
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==LOW && pulsador4==HIGH
&& pulsador5==LOW && pulsador6==LOW)
    {
          comparator=3;
    myDFPlayer.play(3);  //C
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==LOW && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==LOW)
    {
          comparator=4;
    myDFPlayer.play(4);    //D
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==LOW && pulsador4==LOW
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=5;
    myDFPlayer.play(5);    //E
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==LOW && pulsador4==HIGH
&& pulsador5==LOW && pulsador6==LOW)
    {
     comparator=6;
    myDFPlayer.play(6);     //F
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==LOW && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=7;
    myDFPlayer.play(7);  //G
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==LOW && pulsador4==LOW
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=8;
    myDFPlayer.play(8);    //H
      delay(1000);
    }
    if(pulsador1==LOW && pulsador2==HIGH && pulsador3==LOW && pulsador4==HIGH
&& pulsador5==LOW && pulsador6==LOW)
    {
     comparator=9;
    myDFPlayer.play(9);    //I
      delay(1000);
    }
    if(pulsador1==LOW && pulsador2==HIGH && pulsador3==LOW && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=10;
    myDFPlayer.play(10);     //J
```

```
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==HIGH && pulsador4==LOW
&& pulsador5==LOW && pulsador6==LOW)
    {
     comparator=11;
    myDFPlayer.play(11);    //K
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==HIGH && pulsador4==LOW
&& pulsador5==LOW && pulsador6==LOW)
    {
     comparator=12;
    myDFPlayer.play(12);     //L
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==HIGH && pulsador4==HIGH
&& pulsador5==LOW && pulsador6==LOW)
    {
     comparator=13;
    myDFPlayer.play(13);  //M
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==HIGH && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=14;
    myDFPlayer.play(14);   //N
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==LOW && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==HIGH)
    {
     comparator=15;
    myDFPlayer.play(15);  //Ñ
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==HIGH && pulsador4==LOW
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=16;
    myDFPlayer.play(16);  //O
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==HIGH && pulsador4==HIGH
&& pulsador5==LOW && pulsador6==LOW)
    {
     comparator=17;
    myDFPlayer.play(17);  //P
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==HIGH && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=18;
```

```
    myDFPlayer.play(18);   //Q
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==HIGH && pulsador4==LOW
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=19;
    myDFPlayer.play(19);   //R
      delay(1000);
    }
    if(pulsador1==LOW && pulsador2==HIGH && pulsador3==HIGH && pulsador4==HIGH
&& pulsador5==LOW && pulsador6==LOW)
    {
     comparator=20;
    myDFPlayer.play(20);  //S
      delay(1000);
    }
    if(pulsador1==LOW && pulsador2==HIGH && pulsador3==HIGH && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==LOW)
    {
     comparator=21;
    myDFPlayer.play(21);    //T
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==HIGH && pulsador4==LOW
&& pulsador5==LOW && pulsador6==HIGH)
    {
     comparator=22;
    myDFPlayer.play(22);   //U
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==HIGH && pulsador3==HIGH && pulsador4==LOW
&& pulsador5==LOW && pulsador6==HIGH)
    {
     comparator=23;
    myDFPlayer.play(23);    //V
      delay(1000);
    }
    if(pulsador1==LOW && pulsador2==HIGH && pulsador3==LOW && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==HIGH)
    {
     comparator=24;
    myDFPlayer.play(24);    //W
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==HIGH && pulsador4==HIGH
&& pulsador5==LOW && pulsador6==HIGH)
    {
     comparator=25;
    myDFPlayer.play(25);    //X
      delay(1000);
    }
    if(pulsador1==HIGH && pulsador2==LOW && pulsador3==HIGH && pulsador4==HIGH
&& pulsador5==HIGH && pulsador6==HIGH)
    {
```

```
      comparator=26;
     myDFPlayer.play(26);  // Y
       delay(1000);
    }
     if(pulsador1==HIGH && pulsador2==LOW && pulsador3==HIGH && pulsador4==LOW
&& pulsador5==HIGH && pulsador6==HIGH)
    {
      comparator=27;
     myDFPlayer.play(27);  //Z
       delay(1000);
    }

    }
     if(i==1 && pulsador==HIGH && comparator==Nrandom)  //configuracion de las funciones
cuando en el modo juego el usuario acciona una letra correcta
    {

      myDFPlayer.play(28);
      delay(1000);
      myDFPlayer.pause();  //pause the mp3
      i=0;


    }
    if(i==1 && pulsador==HIGH && comparator!=Nrandom)//configuracion de las funciones
cuando en el modo juego el usuario acciona una letra incorrecta
    {
       myDFPlayer.play(29);
       delay(1000);
       myDFPlayer.pause();  //pause the mp3

    }

}
```

CONCLUSIONES Y PROPUESTAS DE MEJORA

A la hora de la realización del proyecto han surgido varios fallos por la calidad de los componentes, los botones al ser de baja calidad si se aplica un poco más de calor del indicado la pletina de metal donde se hacen las conexiones se mueve y da lugar a fallo de los mismos. El DFPlayer Mini a la hora del primer conexionado se me rompió y tuve que comprar otro urgentemente ya que sin él no podía seguir desarrollando el proyecto.
Con la programación me han surgido problemas parecidos al principio en el modo de juego/random al usar el propio comando del DFPlayer Mini no era capaz de hacerlo ya que no daba un valor determinado, hasta que no lo hice con la función random del propio arduino no pude seguir desarrollandolo. Con este modo también me percaté de que siempre seguía la misma secuencia de letras a la hora de usar el modo random, cuando busque sobre esto era en relación a la semilla que se usaba que siempre usaba la misma por defecto y siempre producía la misma secuencia de números "aleatorios" por lo que siempre seguía el mismo orden, por ello en el código esta la linea *randomSeed(analogRead(A0));* ya que gracias a esta se genera una semilla aleatoria gracias a que cada vez que se inicia el arduino se genera un nivel de ruido en las entradas analogicas por lo que se aprovecha esto para generar la semilla en función a este nivel ya que eso sí que será siempre aleatorio.
En cuanto a funcionalidad cuenta con lo pedido y funciona correctamente con los parámetros especificados

Con la implementación de dos botones más se podría usar para la adaptación del lenguaje braille computerizado en el cual es posible la realización de numerosos y diferentes signos.

También se podría implementar un regulador para el volumen ya fuera un potenciómetro o algo por el estilo.

La mejor colocación de los cables y el diseño de una caja especifica para esto ya sea en impresión 3D, madera o plástico. Para facilitar el uso y hacerla más ergonómica.

El diseño de una placa específica donde estuvieran todas las conexiones para ahorrar espacio y dinero.

Un posible indicador de modo en el que fuera posible incluir los códigos de números y mayúsculas para poder añadirle más funcionalidades al proyecto.

WEBGRAFÍA

- https://openwebinars.net/blog/tutorial-arduino-entradas-2-botones/
- https://www.luisllamas.es/arduino-mp3-dfplayer-mini/
- https://programarfacil.com/blog/random-en-arduino-como-usar-numeros-aleatorios/#:~:text=soluci%C3%B3n%20en%20Arduino-,La%20funci%C3%B3n%20Random%20en%20Arduino,aleatorio%E2%80%9D%20desde%20min%20hasta%20max
- https://www.discapnet.es/areas-tematicas/tecnologia-inclusiva/productos-de-apoyo/guias-tecnicas/el-alfabeto-braille
- https://www.electroallweb.com/index.php/2020/07/22/modulo-dfplayer-mini-reproductor-mp3-tutorial-completo/
-
- https://es.wikipedia.org/wiki/Braille_(lectura)
- https://www.somostriodos.com/braille/
- https://www.elisayuste.com/becdot-braille/
- https://www.compartolid.es/braille-computerizado/
- https://www.geekfactory.mx/tienda/modulos-para-desarrollo/modulo-tp4056-cargador-bateria/