Proyecto final de ciclo:

Braille

Alumno: Antonio Manuel Montes Ríos
Curso: S21ME

Indice:

1. Justificación del proyecto. 3
2. Objetivos. 4
3. Desarrollo. 5
4. Conclusiones. 26
5. Lineas de investigación futura. 27
6. Resumen. 27
7. Webgrafia. 28

1. Justificación del proyecto:

- Este proyecto consiste en realizar un robot que reproduce letras para que el usuario tenga que poner la combinación de esa letra en los botones rojos. Después de poner la combinación debe usar el botón de corregir para comprobar si lo ha puesto correctamente. Consta de otro botón para decir letras aleatoriamente.

-El robot consta de dos modos de juego.

-Uno es un modo libre el cual se usa para practicar todas las combinaciones libremente sin la presión de la corrección.

-Y un modo aleatorio o Random el cual consiste en que al pulsar un pulsador les diga una letra del abecedario al azar, y la tarea del usuario que utilice el robot es hacer la combinación de la letra asignada por el modo Random.

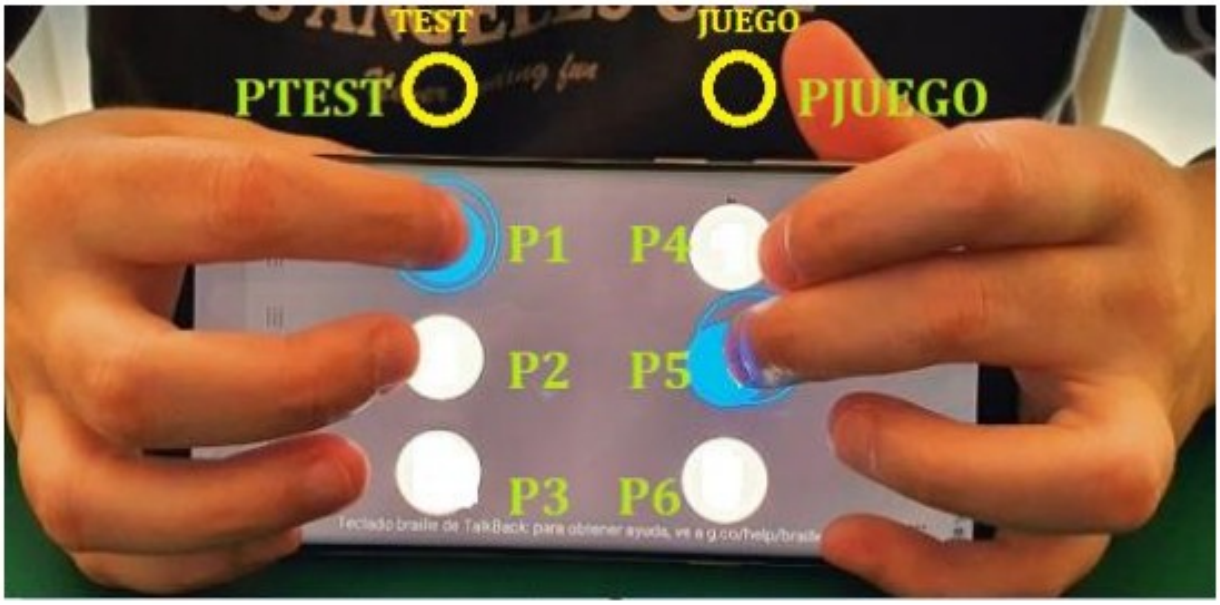
 -Las combinación se realizan usando los interruptores P1, P2, P3, P4, P5 y P6. Los cuales están colocados con la configuración de las letras en braille.

-Al colocar la combinación pulsan del pulsador de corrección para comprobar si es correcto o erróneo.

-En el caso de que use la combinación correcta sonará un check de corrección positiva y en el caso de que la combinación sea errónea sonará una bocina de error indicando que el usuario lo ha colocado mal.

- En el proyecto estamos aplicando todo lo aprendido sobre programación, soldadura y circuitos.

- Todo esto se lleva a cabo usando arduino nano y usando la programación correcta para que haga este objetivo.



2. Objetivos:

- La definición de braille es: El sistema braille o lenguaje para ciegos es el que utilizan las personas con discapacidad visual o ceguera para poder escribir y leer textos, libros y documentos. Se trata de un sistema de lectura y escritura táctil. Su inventor fue Luis Braille (1809-1852) francés nacido el 4 de enero de 1809.

- El sistema Braille utiliza una serie de puntos en relieve que se interpretan como letras del alfabeto y gracias a este método pueden leer y escribir.

- Hemos elegido este proyecto porque parecía muy interesante el poder hacer un juguete que pueda ayudar a las personas invidentes a aprender una de las cosas más importantes para su día a día como es el braille.

- En la exposición que nos dieron las chicas de la once nos enseñaros varios aparatos que se usaban para enseñar este tipo de lenguaje (braille).

- La finalidad del proyecto es que los niños pequeños con esta discapacidad puedan aprender este lenguaje de una forma más sencilla.

3. Desarrollo:

A) Fundamentación teórica:

ABC SOCIEDAD

Buscar

Opinión ▾ España ▾ Economía ▾ Internacional ▾ Sociedad Deportes ▾ Cultura ▾ Historia Ciencia Gente Play ▾ EXCLUSIVO PREMIUM Estilo ▾ Más ☰

«Braillin»: el primer muñeco que enseña el lenguaje de los ciegos

La Organización Nacional de Ciegos (ONCE), la juguetera alicantina Famosa y el Instituto Tecnológico del Juguete presentaron hoy en Madrid a 'Braillín', un muñeco que pretende fomentar la integración social a través del juego y familiarizar a los niños, invidentes o no, con el código Braille.

Este juguete, que puede ser utilizado por cualquier niño o niña, con o sin discapacidad visual, constituye la primera iniciativa mundial de este tipo y tiene como principal finalidad enseñar a los niños a respetar y aceptar la diversidad como un elemento enriquecedor.

La directora de Educación de la ONCE, Mª Luz Laine Mouliaá, explicó que con este muñeco se busca "potenciar aquellas actividades y experiencias que permitan el adecuado desarrollo educativo, afectivo y social de los niños y niñas con discapacidad visual, y contribuir a que el resto de sus compañeros conozcan el código Braille y las pautas básicas en la relación con las personas que tienen este tipo de discapacidad. Y todo ello a través de una actividad muy importante para ellos: el juego".

El muñeco fue creado en Argentina por una profesora que trabaja con niños y niñas con discapacidad visual y consiguió un premio en el Concurso de Investigación Educativa sobre Experiencias Escolares, en la modalidad de material didáctico, que todos los años convoca la ONCE.

A partir del prototipo inicial el Instituto Tecnológico del Juguete, siempre en coordinación con profesionales expertos de la ONCE, desarrolló un estudio de observación en siete colegios, con sesiones de juego en las que participaron casi 50 niños y niñas, de 3 a 10 años, en pequeños grupos en los que siempre había alguno con discapacidad visual, para poder valorar las posibilidades de uso del juguete, adecuación ergonómica, atractivo, formas, colores, etc.

Por otra parte, se realizaron entrevistas en profundidad a profesores y expertos en discapacidad visual, en juego y juguete, pedagogos, psicólogos. . . En total, han participado directamente en la investigación 30 profesionales de distintas disciplinas. De esta forma, se obtuvieron los parámetros necesarios para optimizar el producto desde el punto de vista de la calidad formal, lúdica y pedagógica: se identificaron los tamaños de los diferentes accesorios, texturas adecuadas, contrastes de color más idóneos, elementos más apropiados para la cara y cuerpo del muñeco, presión necesaria que debe ejercerse sobre los puntos de Braille, etc.

Siguiendo todas estas indicaciones, el equipo de I+D de Famosa realizó un nuevo diseño del muñeco, para dar respuesta a todas las necesidades y preferencias del público al que va dirigido, según explicó Felipe Gómez, director de Marketing de la empresa. María Costa, directora del Departamento de Pedagogía de AIJU, subrayó que "Braillín representa un excelente recurso didáctico, de enorme valor para las aulas o para el juego en el hogar.

Además, constituye una novedosa forma de jugar a las muñecas, que aúna aprendizaje y diversión, ya que no existe nada parecido para trabajar en los colegios. No sólo actúa como material integrador e instrumento para el aprendizaje del Braille, sino que refuerza la autoestima infantil y la autonomía personal, estimula el desarrollo sensorial y motor, y permite conocer y trabajar el esquema corporal y las nociones espaciales y matemáticas".

En este sentido, Braillín se considera un material especialmente apropiado para la Educación Infantil y el primer ciclo de Educación Primaria. El muñeco viene acompañado de una Guía Didáctica, en la que se explica la utilidad de Braillín, proponiéndose diferentes actividades para realizar en entornos educativos, un alfabeto Braille en relieve y tinta, un póster de alfabeto Braille y una plantilla de puntos Braille.

En total, la ONCE distribuirá más de 1. 100 muñecos entre sus colegios y equipos específicos de Atención a la Educación Integrada de ciegos y deficientes visuales. Braillín se podrá también adquirir en cualquier centro de la ONCE. Según apunta Vicente Ruiz Martínez, director general adjunto de Servicios Sociales de la ONCE, "más de 7. 500 niños y jóvenes se benefician de las ayudas educativas de la Organización Nacional de Ciegos (ONCE), que cada curso les proporciona los materiales e instrumentos necesarios para la adaptación de sus puestos de estudio y que cuenta con 650 profesionales dedicados a la educación".

Enlace a la noticia: https://www.abc.es/sociedad/abci-braillin-primer-muneco-ensena-lenguaje-ciegos-200411240300-963609552623_noticia.html?ref

B) Materiales y métodos:

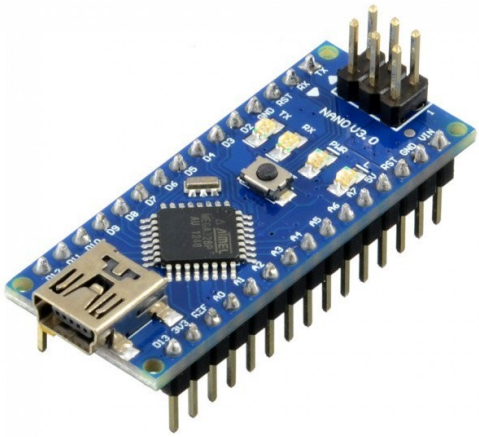
Los materiales usados son los siguientes:

- Arduino Nano.(1)
- DFPlayermini.(2)
- Altavoz.(3)
- Targeta SD.(4)
- 2 Pulsadores amarillos, 6 Interruptores rojos y 1 interruptor de palanca.(5)
- Cable de comunicación USB/MiniUSB.(6)
- Cableado.(7)
- Cargador de batería TP4056.(8)
- Batería.(9)
- Conversor HW-106.(10)
- Conector Jack Hembra stereo 3.5mm Chasis.(11)
- Zócalos.(12)
- Placa perforada.(13)
- Placa protoboard.(14)

- Caracteristicas de los materiales:

1. Arduino Nano

- Microcontrolador Arduino ATmega328Arquitectura
- Arquitectura AVR
- Voltaje de operación 5V
- Memoria flash 32 KB de los cuales 2 KB utilizados por bootloader
- SRAM 2 KB
- Velocidad del reloj 16 MHz
- Pines de E/S analógicas, 8
- EEPROM 1 KB
- Corriente continua por pin entrada salida, 40 mA (Pines de E/S)
- Voltaje de entrada, 7-12 V
- Pines de E/S digitales, 22
- Salida PWM, 6
- Consumo de energía, 19 mA
- Tamaño de la placa de circuito impreso, 18 x 45 mm
- Peso 7 g



2. DFPlayer Mini:

- Alimentación: 3.2V a 5V
- Samples soportados (kHz): 8/11.025/12/16/22.05/24/32/44.1/48
- Salida: 24 -bit DAC con soporte para rango dinámico 90dB
- Sistema de archivos: FAT16 y FAT32. Máx: 32G (tarjeta Micro SD no incluida)
- Interfaz: Pines de control I/O para pulsadores o modo Serial
- Soporta hasta 100 directorios. Cada directorio soporta un máximo de 255 archivos de audio
- Volumen ajustable en 30 niveles

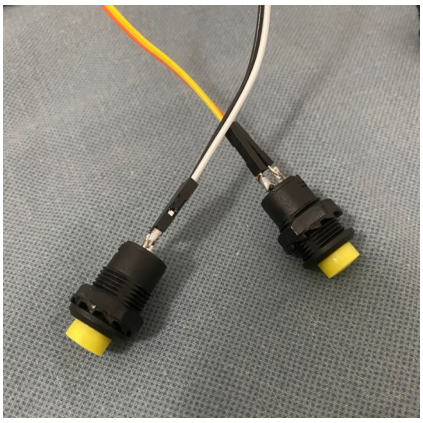
3. Altavoz:

- Diámetro de 65 mm
- Potencia: No especificado
- Resistencia interna de 8 ohmios

4. Tarjeta SD:

Dimensiones: 15 x 110 x 1 milímetros
Peso: 0’258g aprox
Voltaje: 2’7 – 3’6
Número de pines: 8
Protección contra escritura: No
Protectores de terminal: No

5. Pulsadores e interruptores:



6. Cable comunicación USB a mini USB:

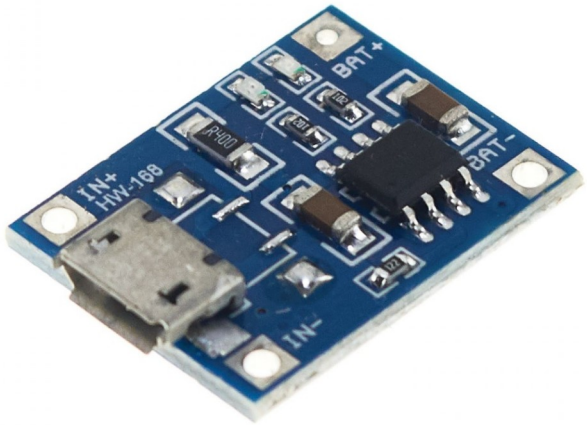
- Velocidad de transferencia de datos hasta 480 Mbps

7. Cableado:

- Longitud: 10cm – 30cm
- Tipo de conexión: Hembra - Hembra, Hembra – Macho, Macho – Macho.

8. Cargador de batería TP4056:

Voltaje de Entrada DC: 4’5V – 5’5V
Corriente de 1A ajustable
Módulo de carga lineal
Precisión de carga: 1’5%
Led indicador color Rojo: Cargando Batería
Led indicador color Verde: Batería con carga completa
Conexión Micro USB
Temperatura de trabajo: -10ºC a 85ºC
Dimensiones: 25mm x 19x 10 mm



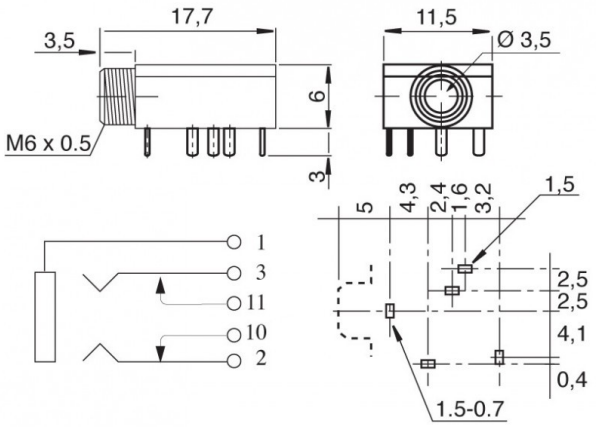
9. Batería:

- Voltaje: 3’7V
- Capacidad: 1200mAh

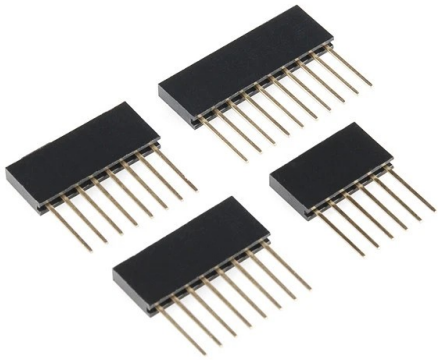
10. Conversor HW-106:

Tamaño: 34mm x 16mm x 10mm
Peso: 4’5g
Rango de voltaje de entrada: 0’9 – 5V
Voltaje de salida: 5V
Corriente de salida: 600 mAh

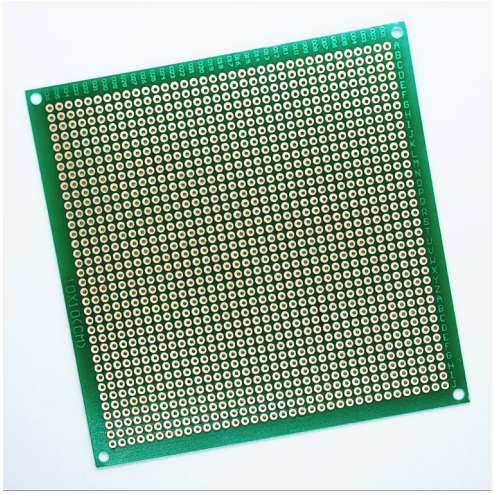
11. Conector Jack Hembra stereo 3.5mm Chasis:



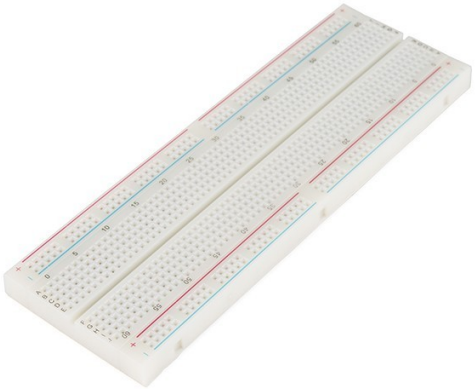
12. Zócalos:



13. Placa perforada:



14. Placa protoboard:



- El procedimiento que se ha seguido para llevarlo a cabo el proyecto es:

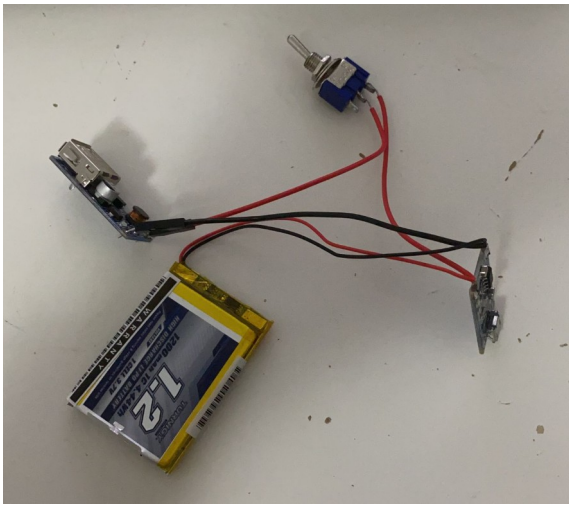
Primero se ha montado todo el circuito y después si ha realizado toda la programación mediante el ID de arduino.

C) Resultados y análisis:

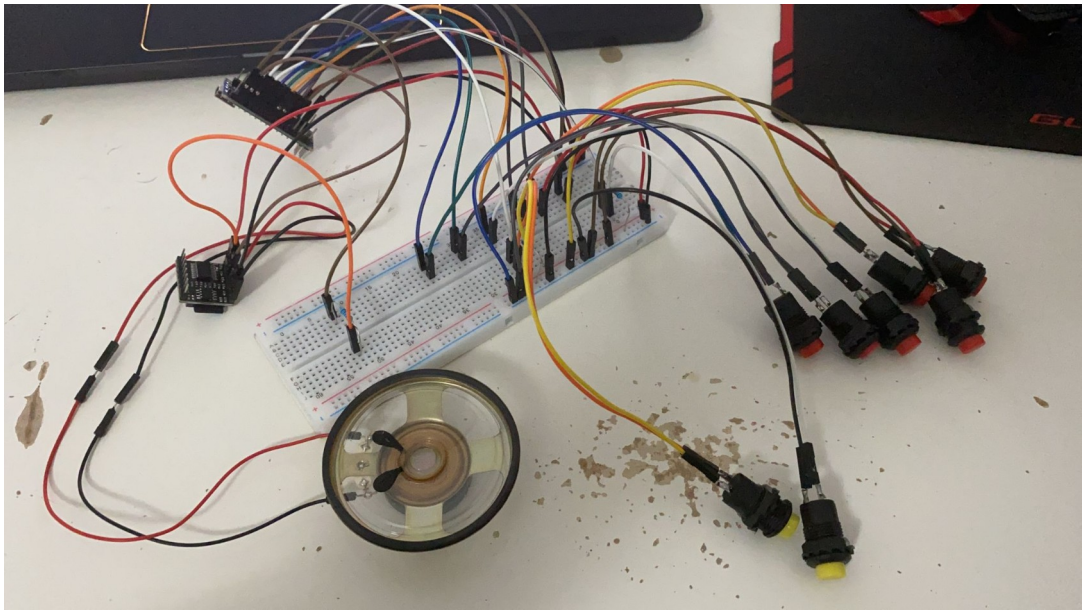
- El resultado del montaje son el chasis, el circuito de alimentación y el circuito del conjunto del arduino, DFPlayermini, altavoz, pulsadores e interruptores.

- Se mostraran todos las partes usadas para montarlo:

- Circuito de alimentación:



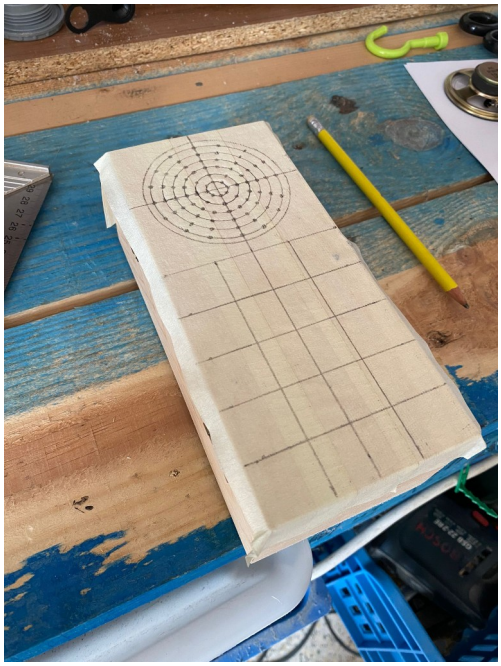
- Circuito del conjunto del arduino, DFPlayermini, altavoz, pulsadores e interruptores:



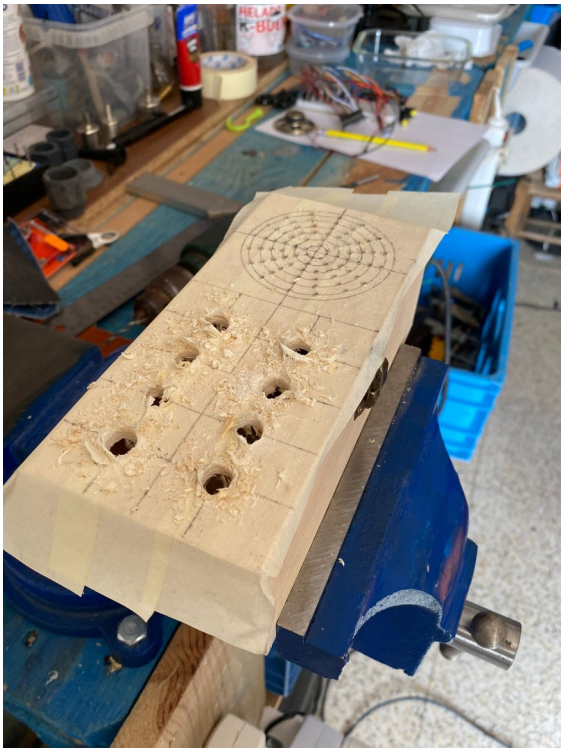
- Pasos seguidos para la fabricación del chasis del robot:

1. Se usa una caja de madera como base para la estructura del robot:

2. Se procede a marcar y trazar las ubicaciones de cada elemento para no cometer fallos:



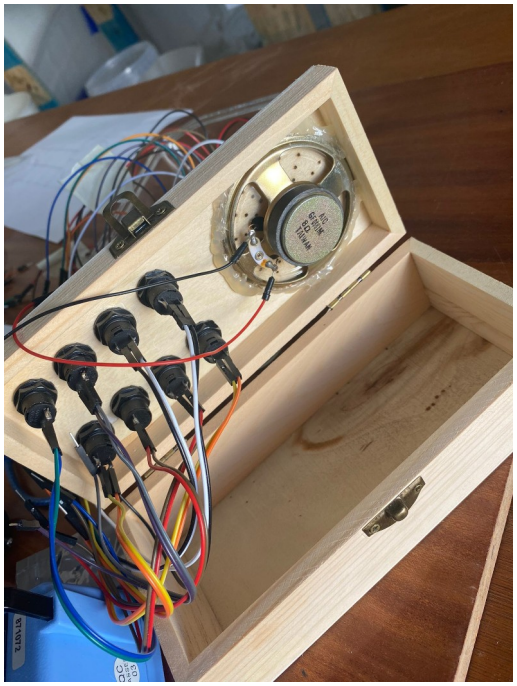
3. Una vez delimitado cada lugar se hacen los orificios necesarios para los botones y los que permiten que salga sonido del altavoz:



4. Tras hacer los orificios se procede a lijar la zona para dejarla lisa:

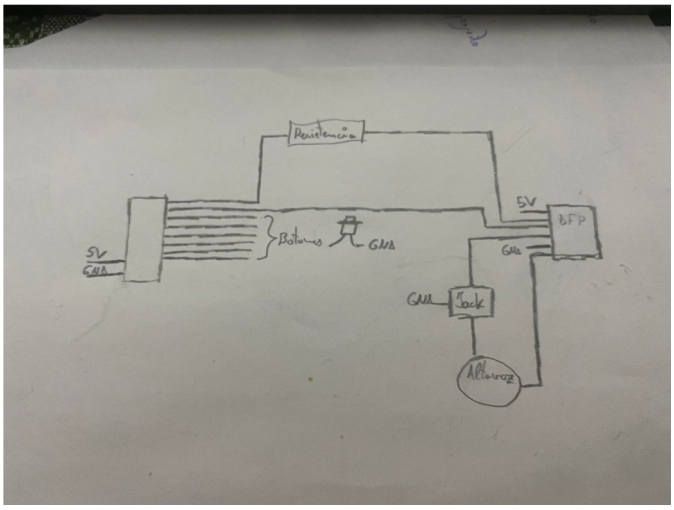
- Montaje del circuito en el chasis:

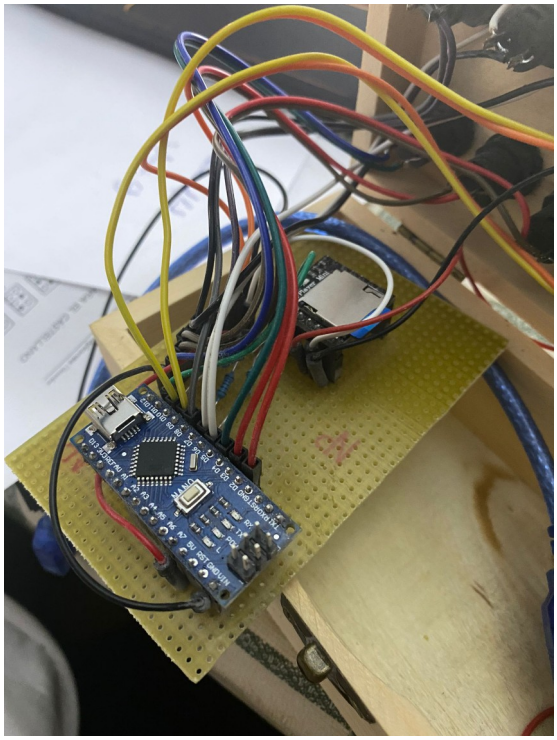
1. Primero monto los botones y el altavoz:

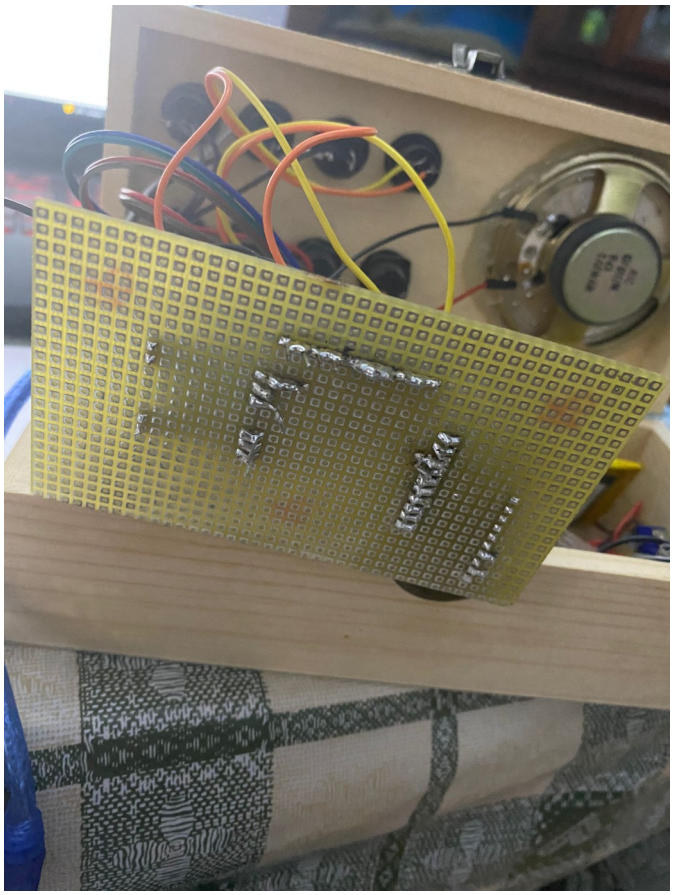


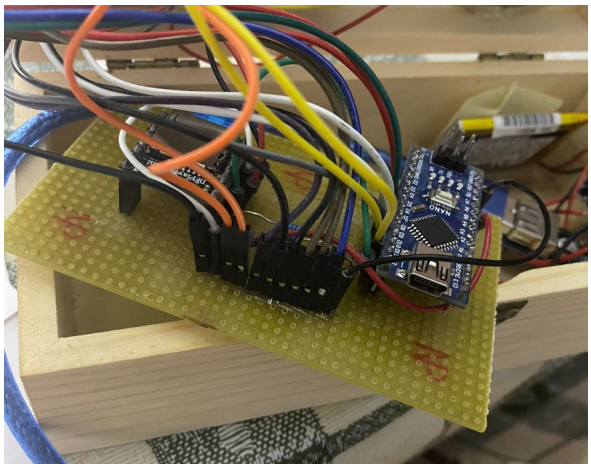
2. Tras comprobar que las conexiones de los botones eran demasiado grandes y no permitían cerrar la caja se desoldó y se corta el conector para eliminar este problema:

- Después de hacer un croquis para tener una guía donde soldar cada componente se suelda primero los zócalos, botones y altavoz, GND y VCC y por ultimo las resistencias necesarias.

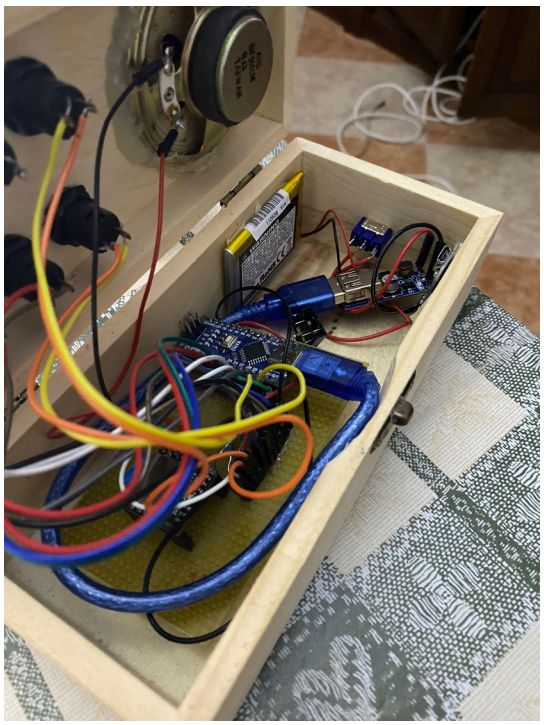


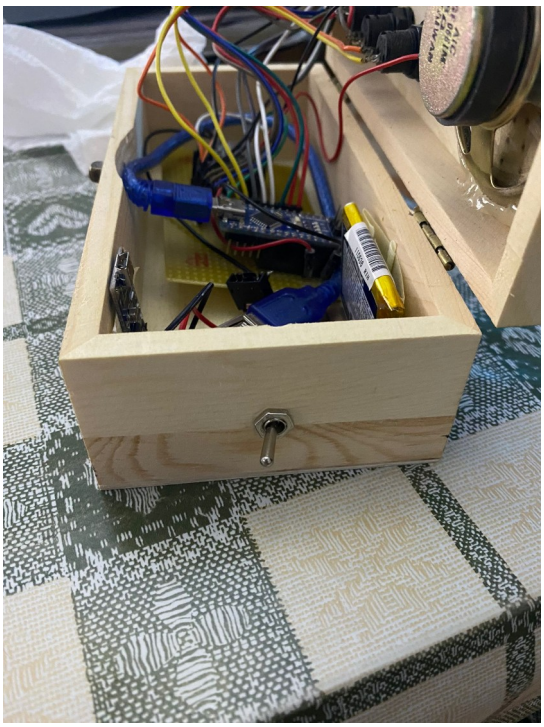




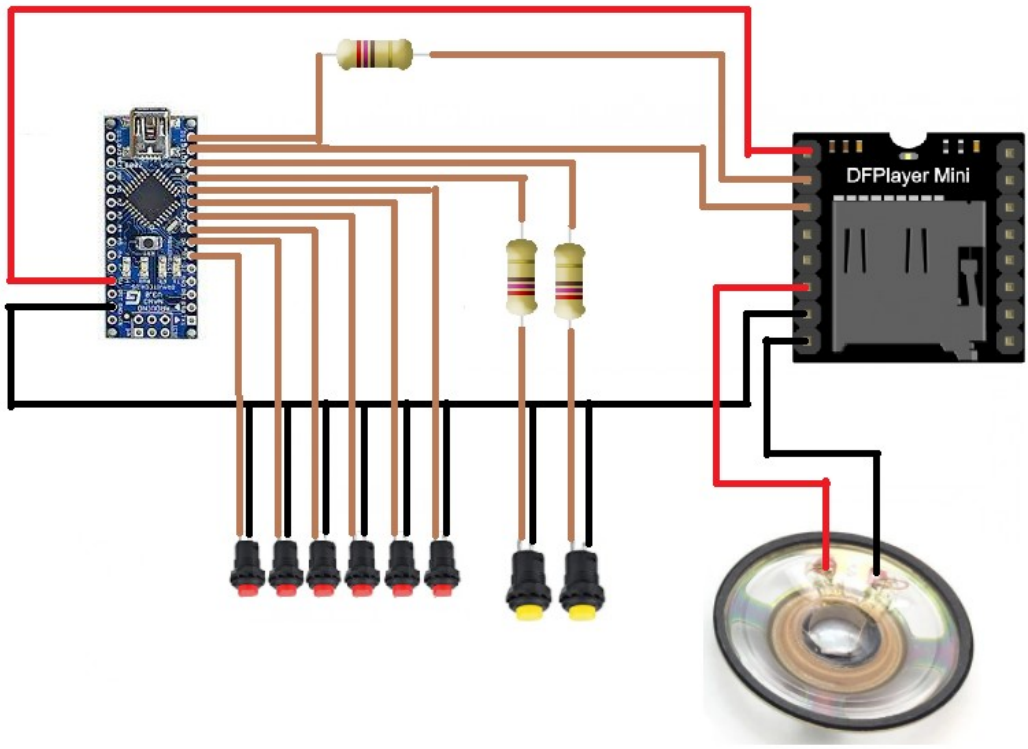


- La ultima parte fue montar la parte de la alimentación, hubo que hacer otro orificio par el interruptor de encendido y apagado, y limar un borde del chasis para que entre el cable de alimentación:





- Esquema del circuito:

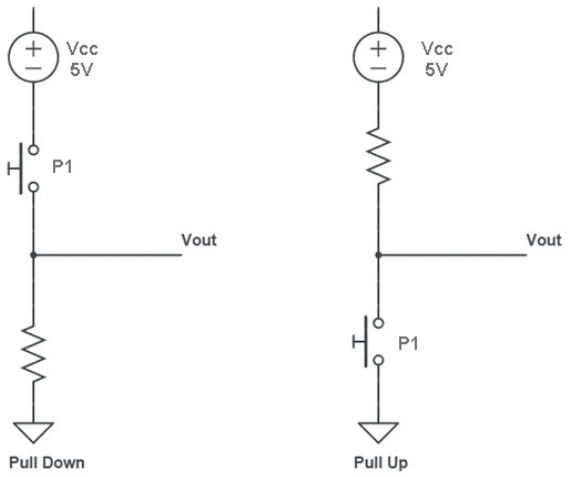


* Explicación de porque los botones rojos no llevan resistencia:

- Los botones rojos no usan resistencia física debido a que en el código está especificado que use resistencias internas del propio arduino.

```
pinMode(s1, INPUT_PULLUP);
pinMode(s2, INPUT_PULLUP);
pinMode(s3, INPUT_PULLUP);
pinMode(s4, INPUT_PULLUP);
pinMode(s5, INPUT_PULLUP);
pinMode(s6, INPUT_PULLUP);
```

- En cambio los botones amarillos usan las resistencias de Pull Up para determinar que en reposo mandan 1 o ``HIGH´´ y al presionarlo manda 0 o ``LOW´´.



- Explicación de la programación por partes: Cada explicación está señalada con un *.

```
#include "Arduino.h "
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"
```

*Aquí se incluyen las librerías necesarias para que todo funcione correctamente.

```
//Pulsadores

const int s1 = 2; // pulsadores
const int s2 = 3;
const int s3 = 4;
const int s4 = 5;
const int s5 = 6;
const int s6 = 7;
const int resuelve = 8;
const int aleatorio = 9;
```

* En esta parte se indica con que pin corresponde pulsador.

```
int boton_aleatorio = 0;
int resolviendo = 0;
int boton1 = 0;
int boton2 = 0;
int boton3 = 0;
int boton4 = 0;
int boton5 = 0;
int boton6 = 0;

int numL = 0;

int digito = 0;

int randomNumber = 0;
```

* Los int almacenan valores numéricos en la variable indicada.

```
SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);
```

* Esta libreria permitir la comunicación en serie por otros pines digitales del Arduino, usando un software para replicar la misma funcionalidad.

```
boolean Juego = false;
```

* Esta es una función que almacena valores binarios 0 o 1. Y lo representa como TRUE o FALSE.

```
void setup()
{
  mySoftwareSerial.begin(9600);
  Serial.begin(115200);
  if (!myDFPlayer.begin(mySoftwareSerial)) {  //Use softwareSerial to communicate with mp3.
  }
  myDFPlayer.setTimeOut(500); //Set serial communictaion time out 500ms
  //----Set volume----
  myDFPlayer.volume(20);  //Set volume value (0~30).
  myDFPlayer.volumeUp(); //Volume Up
  myDFPlayer.volumeDown(); //Volume Down
  //----Set different EQ----
  myDFPlayer.EQ(DFPLAYER_EQ_NORMAL);
  //----Set device we use SD as default----
  myDFPlayer.outputDevice(DFPLAYER_DEVICE_SD);
```

* Aquí está la configuración del DFPlayer.

```
  pinMode(s1, INPUT_PULLUP);
  pinMode(s2, INPUT_PULLUP);
  pinMode(s3, INPUT_PULLUP);
  pinMode(s4, INPUT_PULLUP);
  pinMode(s5, INPUT_PULLUP);
  pinMode(s6, INPUT_PULLUP);
```

* Esta parte es la que indica que los pulsadores s1, s2, s3, s4, s5, y s6 usan resistencias internas del arduino.

```
}
void loop()
{
  boton1 = digitalRead(s1);          // El estado 1 es igual al modo del interruptor1
  boton2 = digitalRead(s2);          // El estado 2 es igual al modo del interruptor2
  boton3 = digitalRead(s3);
  boton4 = digitalRead(s4);
  boton5 = digitalRead(s5);
  boton6 = digitalRead(s6);
  resolviendo = digitalRead(resuelve); // La variable resolviendo es igual al modo del pulsador A
  boton_aleatorio = digitalRead(aleatorio);
```

* Aquí se relacionan los nombres de cada boton con la lectura digital que hace cada pin.

```
 if (resolviendo == LOW && Juego == 0) {
    Juego = 0;
      numL = numeroLetra();
      if ( numL != 0 ) {
      myDFPlayer.play(numL);    // Suena Letra indicada
      delay(1000);          // descansa 1 segundo.
                    }
                       }
```

* Este IF es el que permite que sin estar en el modo aleatorio se pueda hacer las combinaciones de cada letra y corregirlas. Indica que si el botón de corregir es igual a LOW ``presionado´´ y la función boleana es igual a 0 (significa que el modo aleatorio no está activado) el DFPlayer dice la combinación puesta por el usuario y solo dice la letra.

```
       if( boton_aleatorio == LOW) {
            Juego = 1;
           randomNumber = random(1,26);

           myDFPlayer.play(randomNumber);
           delay(1000);          // descansa 1 segundo.
                              }
           if (resolviendo == LOW && Juego == 1)
           {  numL = numeroLetra();
             if( numL == randomNumber ){
               myDFPlayer.play(numL);
               delay(1000);
               myDFPlayer.play(27);
               delay(1000);
               Juego = 0;
             }
```

```
        else {
            myDFPlayer.play(numL);
            delay(1000);
            myDFPlayer.play(28);
            delay(1000);
            Juego = 0;
        }
      }
```

* Este IF es el que permite hacer el modo aleatorio, lo que indica es que si el botón aleatorio está en LOW la función boleana de juego pasa a esta en 1 indicando que se corregirá la palabra que ponga, después hace que el DFPlayer diga un número aleatorio con el randomNumber, a continuación indica que si el botón de corregir está en LOW ``presionado´´ y el juego está en 1 compara que la letra dicha por el modo aleatorio y la combinación puesta por el usuario, si es la misma lo corrige con un check positivo y si no concuerdan la corrige con una bocina negativa.

```
 }
  int numeroLetra()
```

* Aquí se incluye una variable para que almacene la combinación colocada en los pulsadores.

```
{
   if (boton1 == HIGH && boton2 == LOW && boton3 == LOW && boton4 == LOW && boton5 == LOW && boton6 == LOW)
   {
    return 1;   //A
   }

   else if (boton1 == HIGH && boton2 == HIGH && boton3 == LOW && boton4 == LOW && boton5 == LOW && boton6 == LOW)
   {
    return 2;   //B
   }

   else if (boton1 == HIGH && boton2 == LOW && boton3 == LOW && boton4 == HIGH && boton5 == LOW && boton6 == LOW)
   {
    return 3;   //C
   }

   else if (boton1 == HIGH && boton2 == LOW && boton3 == LOW && boton4 == HIGH && boton5 == HIGH && boton6 == LOW)
   {
    return 4;   //D
   }

   else if (boton1 == HIGH && boton2 == LOW && boton3 == LOW && boton4 == LOW && boton5 == HIGH && boton6 == LOW)
   {
    return 5;   //E
   }
```

```
  else if (boton1 == HIGH && boton2 == HIGH && boton3 == LOW && boton4 == HIGH && boton5 ==
LOW && boton6 == LOW)
  {
    return 6;   //F
  }

  else if (boton1 == HIGH && boton2 == HIGH && boton3 == LOW && boton4 == HIGH && boton5 ==
HIGH && boton6 == LOW)
  {
    return 7;   //G
  }

  else if (boton1 == HIGH && boton2 == HIGH && boton3 == LOW && boton4 == LOW && boton5 ==
HIGH && boton6 == LOW)
  {
    return 8;   //H
  }

  else if (boton1 == LOW && boton2 == HIGH && boton3 == LOW && boton4 == HIGH && boton5 ==
LOW && boton6 == LOW)
  {
    return 9;   //I
  }

  else if (boton1 == LOW && boton2 == HIGH && boton3 == LOW && boton4 == HIGH && boton5 ==
HIGH && boton6 == LOW)
  {
    return 10;   //J
  }

  else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == LOW && boton5 ==
LOW && boton6 == LOW)
  {
    return 11;   //K
  }

  else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == LOW && boton5 ==
LOW && boton6 == LOW)
  {
    return 12;   //L
  }

  else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == HIGH && boton5 ==
LOW && boton6 == LOW)
  {
    return 13;   //M
  }

  else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == HIGH && boton5 ==
HIGH && boton6 == LOW)
  {
    return 14;   //N
  }

  else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == LOW && boton5 ==
HIGH && boton6 == LOW)
  {
```

```
    return 15;   //O
  }

  else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == HIGH && boton5
== LOW && boton6 == LOW)
  {
    return 16;   //P
  }

  else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == HIGH && boton5
== HIGH && boton6 == LOW)
  {
    return 17;   //Q
  }

  else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == LOW && boton5 ==
HIGH && boton6 == LOW)
  {
    return 18;   //R
  }

  else if (boton1 == LOW && boton2 == HIGH && boton3 == HIGH && boton4 == HIGH && boton5 ==
LOW && boton6 == LOW)
  {
    return 19;   //S
  }

  else if (boton1 == LOW && boton2 == HIGH && boton3 == HIGH && boton4 == HIGH && boton5 ==
HIGH && boton6 == LOW)
  {
    return 20;   //T
  }

  else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == LOW && boton5 ==
LOW && boton6 == HIGH)
  {
    return 21;   //U
  }

  else if (boton1 == HIGH && boton2 == HIGH && boton3 == HIGH && boton4 == LOW && boton5 ==
LOW && boton6 == HIGH)
  {
    return 22;   //V
  }

  else if (boton1 == LOW && boton2 == HIGH && boton3 == LOW && boton4 == HIGH && boton5 ==
HIGH && boton6 == HIGH)
  {
    return 23;   //W
  }
  else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == HIGH && boton5 ==
LOW && boton6 == HIGH)
  {
    return 24;   //X
  }
```

```
   else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == HIGH && boton5 ==
HIGH && boton6 == HIGH)
   {
     return 25;   //Y
   }

   else if (boton1 == HIGH && boton2 == LOW && boton3 == HIGH && boton4 == LOW && boton5 ==
HIGH && boton6 == HIGH )
   {
     return 26;   //Z
   }

   else
  {
     return 0;
       }

 }
```

* En la parte final del código se indican todas las combinaciones de cada letra indicando desde la A a la Z, el funcionamiento es que va pasando de palabra en palabra hasta encontrar la que concuerde y manda el número del audio a la variable NumeroLetra().

4. Conclusiones:

- Finalmente se ha conseguido realizar el objetivo del proyecto a pesar de algunas complicaciones.

- En mi caso en la parte física de no he tenido ningún problema salvo que se a roto algunos cables de los pulsadores al intentar cerrar la caja, pero conseguí colocarlos en una posición en la cual no se obstruye con ningún componente.

- En cambio la parte de la programación me dio varios fallos.

- En una ocasión me dio fallos la tarjeta SD con los audios, ya que los audios están nombrados de la A a la Z de 1 a 26 y cuando usaba la combinación de la A el DFPlayer usaba el audio de otra letra, la solución que encontré a este problema fue formatear varias veces la tarjeta hasta conseguir que funcionara correctamente, ya que cada vez que la formateaba cambia el orden de los audios a pesar de que los audios estaban nombrados correctamente.

- Otros de los fallos que me estuvo dando la programación fue que el modo random no funcionaba como debía a pesar de que el código estaba correcto, y estaba correcto porque al siguiente día fui a probarlo de nuevo y empezó a funcionar correctamente sin haber cambiado nada del código ni del circuito.

- Otro de los problemas que tuve fue que la soldar los componentes en la placa perforada no pensé en el cable USB a mini USB, lo cual hizo que no entrara a lo ancho del chasis. Pero con lijé el filo del chasis donde chocaba el cable y conseguí que entrara correctamente sin que choque con ningún cable ni componente.

5. Lineas de investigación futuras:

- El proyecto ha sido terminado con su objetivo principal, pero si pudiéramos seguir avanzando con el robot creo que deberíamos informarnos sobre las necesidades que tengan las personas con esta discapacidad, y en base a eso avanzar por un sitio o por otro.

- Una de las ampliaciones que he pensado es en hacer un código el cual vaya guardado las letras que va colocando el usuario y al final poder formar una palabra. Por ejemplo para vocabularios de algún tipo, como de animales o de colores, esto se puede aplicar a cualquier ámbito.

- Otra de las ampliaciones que he pensado sería un rediseño. Este rediseño constaría solo de un pulsador y cambiaría los interruptores rojos por servomotores que formen las letras, así solamente tendrían que pulsar un botón y los servos colocarían una letra al azar en vez de formarla con los botones. Este proyecto he pensado que podría ser para gente con mas experiencia en el braille que tengan más nociones de todas las palabras.

6. Resumen:

- Este proyecto final consiste en realizar un robot mediante arduino en colaboración con la once, para enseñar a las personas invidentes (principalmente niñ@s pequeños) él lenguaje del braille.

-El robot que hemos realizado se basa en un juguete que nos enseñaron las chicas de la once en una exposición dada en el instituto con el cual los niñ@s aprendían y se divertían a la vez.

-La estructura del robot es de una caja con 8 botones en la cara superior, 6 de los cuáles son para hacer las combinaciones de cada una de las letras, los otros dos son uno para el modo random o aleatorio y el otro para corregir las palabras.

-Este proyecto representa una forma de aprendizaje interactiva muy interesante para los niños pequeños, tanto invidentes como para niños sin esta discapacidad al reproducir sonidos puede llamar su atención y con ello que se interesen por este lenguaje.

-Una parte buena que tiene el proyecto es que se puede poner en cualquier idioma simplemente cambiando los audios de las tarjetas SD, solo hay que nombrarlos con su letra correspondiente y además lo puedes configurar a tu gusto sustituyendo los sonidos de corrección y de error por algunos que hagan sentir más cómodo al usuario que lo use.

7. Webgrafía:

- https://www.abc.es/sociedad/abci-braillin-primer-muneco-ensena-lenguaje-ciegos-200411240300-963609552623_noticia.html?ref=https%3A%2F%2Fwww.google.com%2F
- https://www.hwlibre.com/arduino-nano/
- https://descubrearduino.com/arduino-nano-pinout/
- https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299
- https://www.luisllamas.es/arduino-mp3-dfplayer-mini/
- https://mielectronicafacil.com/proyectos/arduino/maquina-de-sonidos/#page-content
- https://programarfacil.com/blog/arduino-blog/resistencia-pull-up-y-pull-down/
- https://www.makerelectronico.com/producto/dfplayer-mini-reproductor-mp3/
- https://hardzone.es/2019/06/15/cables-usb-guia-diferenciarlos/
- https://ferretronica.com/products/modulo-cargador-baterias-litio-tp4056-micro-usb-5v-1a
- https://arduino.en.alibaba.com/product/62227822291-813084145/HW_106_DC_DC_USB_boost_voltage_regulator_module_step_up_Boost_Adjustable_Voltage_Charger_Power_Supply_Module.html
- https://aprendiendoarduino.wordpress.com/2016/11/16/funciones-definidas-por-usuario-2/
- https://aprendiendoarduino.wordpress.com/2017/06/20/variables-en-arduino/
- https://www.arduino.cc/reference/es/language/variables/data-types/boolean/