Proyecto i-Bray

Jaime Hidalgo Vilaseca

S21ME

23/06/2021

Índice

INTRODUCCION 1
DESARROLLO 3
SIMULACION 3
MATERIALES 4
CIRCUITO 12
CODIGO 14
DIAGRAMA DE BLOQUES 20
CARCASA 21
PRESUPUESTO DE PROTOTIPO 22
CONCLUSIONES 23
AMPLIACIONES 23
WEBGRAFÍA 24
ANEXOS 25

INTRODUCCION

Este proyecto busca crear el prototipo de una herramienta, camuflada en un juguete electrónico, con la que poder difundir el conocimiento del alfabeto Braille para cualquier persona. Sin embargo, el público objetivo principal serán los niños con ceguera en su primera etapa de estudio.

Tras una reunión con el profesorado dedicado a esta especialidad nos hicieron saber que las herramientas/juguetes con los que cuentan para captar la atención y a la vez enseñar a los niños el sistema Braille son muy sencillas, poco interactivas con los niños y muy alejadas de la situación actual en lo que se refiere a la tecnología y la digitalización.

Esto provoca varios inconvenientes para el uso de dichos juguetes:

- Algunas veces los niños se centran más en el propio juguete que en el aprendizaje como por ejemplo con “Braillin” (el muñeco que vemos en la parte superior).
- El juguete por sí solo no interactúa con el alumno por lo que sin la supervisión y guía del profesor no es probable que el alumno avance en su conocimiento.

Para subsanar estos problemas deberíamos crear un juguete cuya función principal sea el sistema Braille, sin olvidar que debe ser atractivo para un niño, además de tener capacidad de interactuación aprovechando las opciones que nos provee la electrónica y tecnología actuales.

El punto de partida para nuestro prototipo procede del proyecto “*Braillear*”, que desarrollaron cinco estudiantes para su tesis de Diseño Industrial en la Universidad de Buenos Aires.

Braillear desarrolló un teclado que sirve para el aprendizaje de letras y números, pero también para escribir textos largos, permitiendo, por ejemplo, tomar de apuntes. La impresora posibilita imprimir en braille y traducir textos convencionales.

Aunque nuestro proyecto tiene un objetivo docente y este busca la conectividad y comunicación se tomó el diseño del teclado como concepto para comenzar el desarrollo del nuestro.

A partir de la información recabada se llegó a la idea de:

- Un juego con seis botones que simularan los puntos Braille.
- Dos pulsadores adicionales, uno para que el juego lea la combinación de los seis botones y devuelva la letra en forma de audio. El otro pulsador entraría en un modo juego donde interactuaría directamente con el alumno pidiéndole que introduzca la combinación de una letra aleatoria y señalando si la combinación es correcta o errónea, mediante señales de audio también.

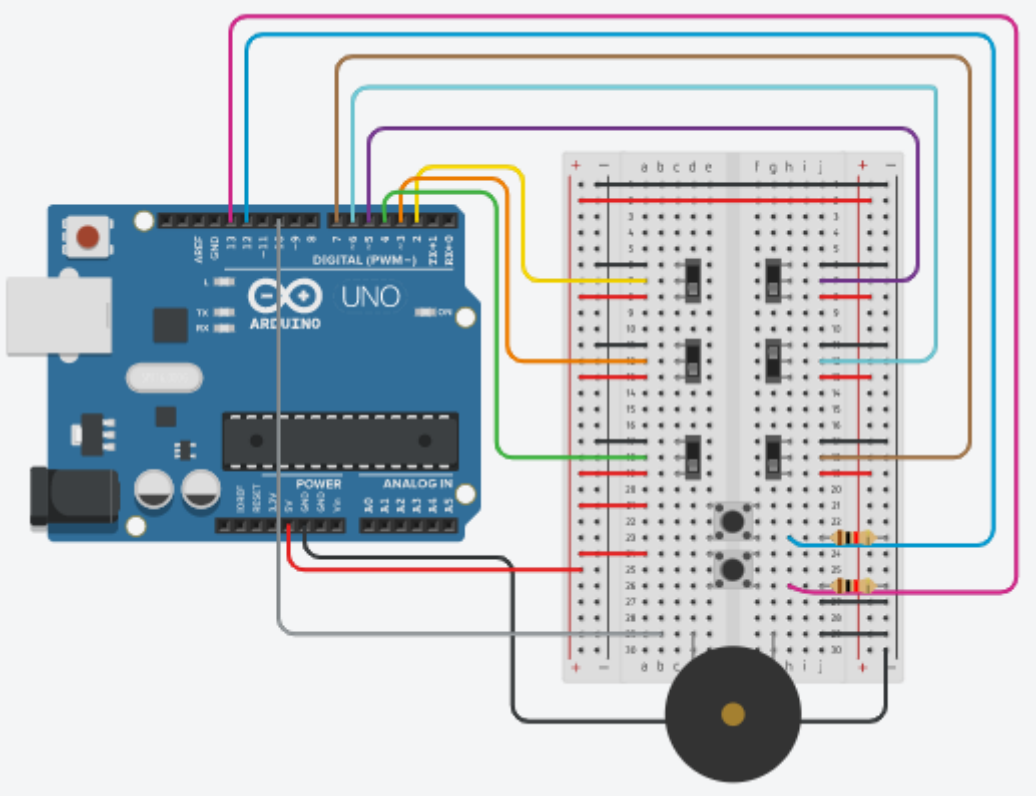
La motivación por la que se ha elegido este proyecto sin duda es la posibilidad de aplicar los conocimientos adquiridos durante el ciclo en la creación de una herramienta que ayude al personal docente en la enseñanza del alfabeto Braille para niños con discapacidad visual.

DESARROLLO

SIMULACION

El primer paso para comprobar la viabilidad del proyecto fue realizar una simulación lo más sencilla posible de la idea del proyecto descrita en la introducción.

Se realizó un montaje en la aplicación web "TINKERCAD" con los seis interruptores que simulan las letras Braille, los dos pulsadores con los que "leer y jugar" y por ultimo un buzzer que cumpliría una función similar al reproductor mp3.



Para la simulación en el código, asigne a cada letra su combinación de pulsadores y las devolvía mediante "serial.print" al accionar el pulsador de lectura. A continuación, al accionar el pulsador de juego se obtenía un numero aleatorio que imprime la letra correspondiente en el alfabeto a dicho número, y por ultimo aumentaba el valor de la variable "MODO" a 1.

Por ultimo añadí un bucle "while" donde permanecer mientras el valor de "MODO" fuera 1. En este bucle introduje una a una las opciones de error y acierto para la combinación requerida por el juego y la introducida por el usuario al accionar el pulsador de lectura.

El código funcionaba correctamente pero no estaba optimizado y se extendía a casi las 400 líneas, por lo que no he visto necesario añadirlo a este documento. Sin embargo, su estructura se mantiene en el código definitivo que desarrollé y que explicaré más adelante.

Tras las pruebas realizadas satisfactoriamente en simulación pasamos al desarrollo del prototipo.

Código completo en Anexo I.

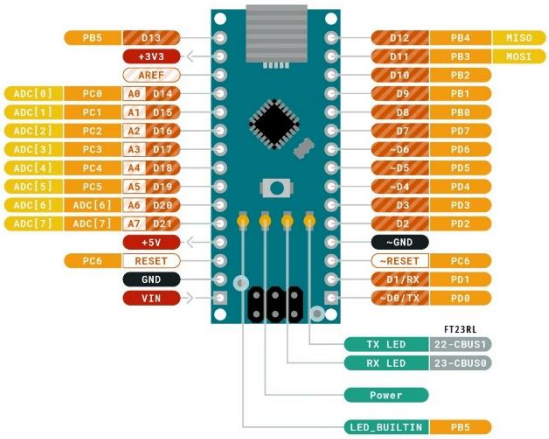
MATERIALES

Microcontrolador

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

En el caso de nuestro proyecto usaremos una versión genérica del Arduino NANO con las siguientes características técnicas:

- Está basada en el microcontrolador o MCU Atmel ATmega328p en versiones 3.x y trabaja a una frecuencia de 16 Mhz.
- La memoria se compone de 16 KB o 32 KB flash según versión (2KB usados para el cargador de arranque), con 1 o 2 KB de memoria SRAM y una EEPROM de 512 bytes o 1 KB según el MCU.
- Tiene una tensión de alimentación de 5v, pero la tensión de entrada puede variar de 7 a 12v.
- Posee 14 pines digitales, 8 analógicos, 2 pines de reinicio y 6 pines de potencia (Vcc y GND). De los pines analógicos y digitales, tienen asignadas varias funciones extra como pinMode() y digitalWrite() y analogRead() para los analógicos. En el caso de los analógicos, permiten una resolución de 10 bits desde 0 a 5v. En los digitales, 22 se pueden usar como salidas PWM.
- Usa un miniUSB estándar para su conexión con el computador para programarla o alimentarla.
- Su consumo de energía es de 19mA.
- El tamaño de la PCB es de 18x45mm con un peso de solo 7 gramos.



Módulo MP3

Arduino no tiene potencia suficiente para reproducir un archivo comprimido como un mp3.

Existen distintas opciones para reproducir ficheros mp3 desde Arduino, como diversos shields y placas. Entre las distintas opciones disponibles para el proyecto, tomaremos el módulo DFPlayer Mini que es un reproductor de audio completo, capaz de reproducir formatos de fichero MP3, WMA y WAV.

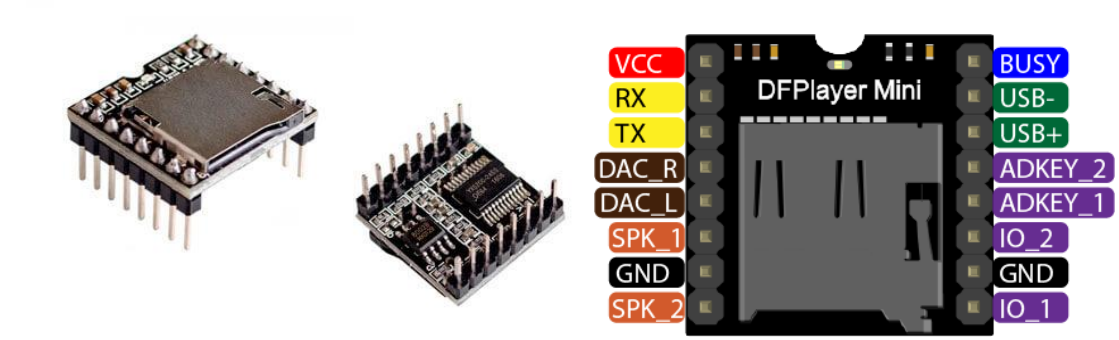
Dispone de un lector micro SD compatible con FAT16 y FAT32, con una capacidad máxima de 32GB. Soporta hasta 100 carpetas y puede acceder hasta 255 canciones.

El DFPlayer Mini proporciona velocidades de muestreo de 8, 11.025, 12, 16, 22.05, 24, 32, 44.1 y 48 kHz, y salida con DAC de 24 bits. Dispone de 30 niveles de volumen ajustable, ecualizador de 6 niveles, y una relación señal ruido (SNR) de 85dB

La comunicación con Arduino se realiza a través de puerto serie. Dispone de funciones para reproducir, detener, pausar, avanzar, retroceder entre canciones. La salida se realiza directamente al altavoz, a través de los pines correspondientes.

características técnicas:

- Dimensiones: 21 mm x 21 mm x 12 mm
- Voltaje de alimentación mínimo: 3.3 V
- Voltaje de alimentación máximo: 5 V
- Admite tasas de muestreo (KHz): 8/11.025/12/16/22,05/24/32/44,1/48
- Salida DAC: 24 bits
- Rango dinámico: 90dB
- Soporte SNR: 85dB
- Compatibilidad: FAT16, sistema de archivos FAT32.
- MicroSD: Si, soporta hasta 32GB.
- Altavoz: No exceder los 3W
- El nombre de los archivos debe incluir una numeración de 4 dígitos.
- Led indicador: Si, cuando comienza a reproducir audio el led prenderá de lo contrario estará apagado



Cargador de batería

Para evitar la necesidad de tener que cambiar la batería cada vez que esta se agote hemos optado por incluir un módulo cargador. Este circuito permite conectarle una fuente de energía eléctrica a su entrada y una batería a su salida para que pueda cargarse de forma adecuada.

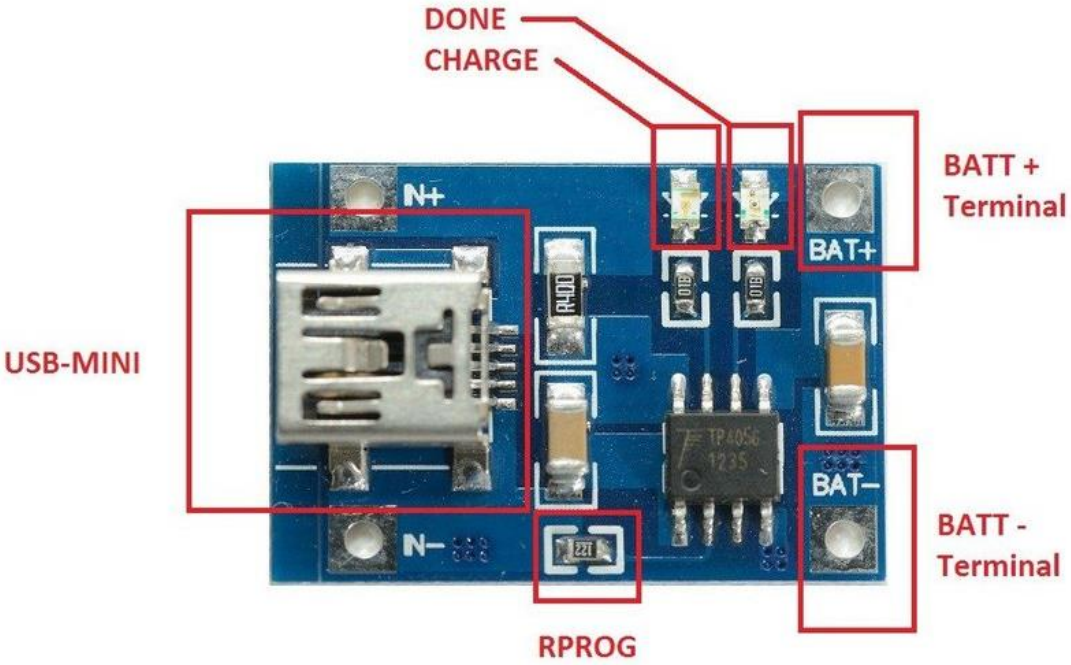
El TP4056 es un chip encapsulado en formato SOP-8 que es capaz de gestionar la carga de una batería. Es decir, adecua la entrada de energía para el estándar de 1A de la mayoría de baterías de litio que se usan en la industria electrónica, y también es capaz de controlar la temperatura.

A través del puerto miniUSB alimentaremos la batería con este tipo de cables. La batería a cargar se conecta a los bornes BAT(+) y BAT(-).

Sus dos LEDs de Carga y de Completado avisan de cuándo la batería está cargándose o cuándo ha terminado el proceso.

Características técnicas:

- Voltaje de entrada: 5V
- Corriente de carga máxima: 1000 mA
- Voltaje de corte de carga: 4.2 V + / – 1%
- Voltaje de protección de sobrecarga de la batería: 2.5 V
- Corriente de protección contra sobre corriente de la batería: 3 A
- Interfaz de entrada: Micro USB
- Dimensión: 2.6 x 1.7 cm

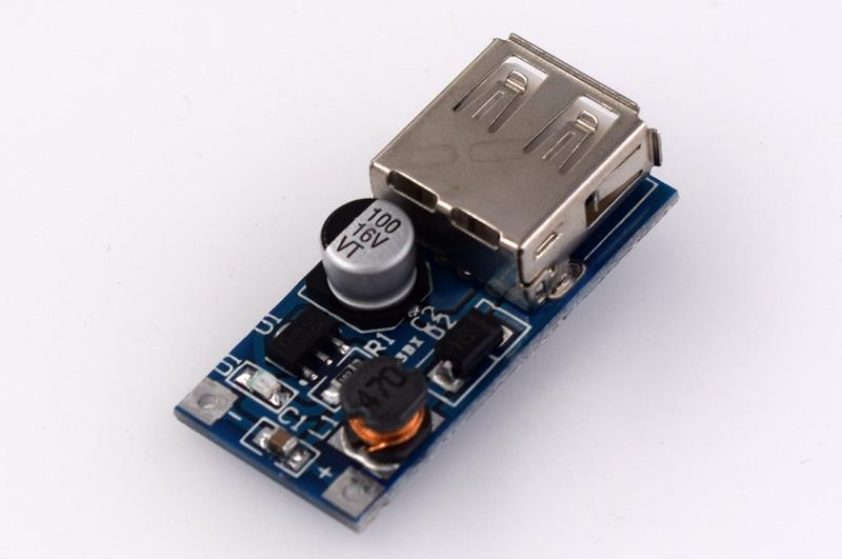


Convertidor DC/DC

El convertidor elevador o convertidor boost es un circuito convertidor de CC a CC que obtiene a su salida una tensión continua mayor que a su entrada. Es un tipo de fuente de alimentación conmutada que contiene al menos dos interruptores semiconductores (diodo y transistor), y al menos un elemento para almacenar energía (condensador, bobina o combinación de ambos). Frecuentemente se añaden filtros construidos con inductores y condensadores para mejorar el rendimiento.

Para nuestro proyecto necesitamos elevar el voltaje entregado por la batería de 3’7V a los 5V necesarios para que el Arduino NANO trabaje correctamente. Hemos elegido el módulo HW-106 que cuenta con las siguientes características técnicas:

- Tamaño del producto 34*16*10mm
- Peso del producto 4,5g
- Rango de tensión de entrada 0,9-5V
- Tensión de salida 5V
- Corriente de salida 600ma



Altavoz

Un altavoz es un transductor electro acústico utilizado para la reproducción de sonido.

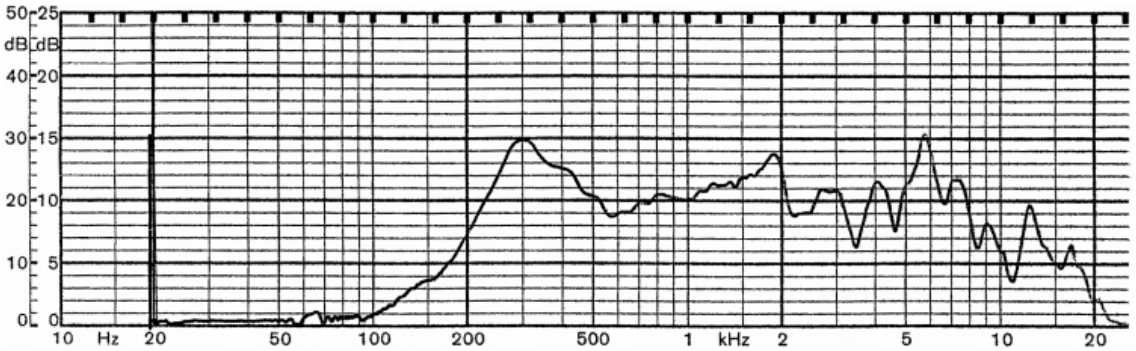
Para nuestro proyecto estamos limitados por la potencia máxima que es capaz de soportar el reproductor DFPLAYER y por el tamaño de nuestro dispositivo.

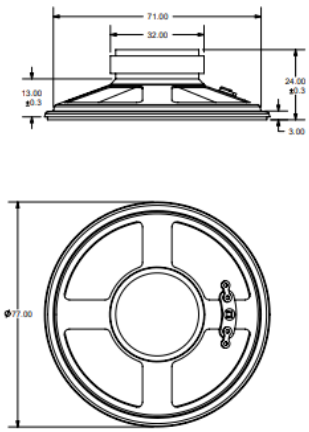
Finalmente elegimos el GF0771M de la marca CUI Devices con las siguientes características técnicas:

- Tamaño/Dimensión - 3.032" D (77.00mm)
- Altura - Asiento (Máx.) - 0.945" (24.00mm)
- Impedancia - 8Ω
- Potencia - Nominal 1 W
- Potencia - Máx. 2 W
- Rango de frecuencia - 280 Hz ~ 11.0 kHz
- Frecuencia de resonancia - 280Hz
- SPL - 90 dB

FREQUENCY RESPONSE CURVE

parameter	conditions/description
potentiometer range	50 dB
rectifier	RMS
lower limit frequency	20 Hz
wr. speed	100 mm/sec
zero level	70 dB





Batería

Las baterías de iones de litio son baterías recargables, en las cuales los iones de litio se mueven entre el ánodo y el cátodo, utilizando un compuesto de litio intercalado como material del electrodo, en lugar del litio metálico, empleado en baterías de litio no recargables.

Para el proyecto elegimos una batería Turnigy 1200mAh 1S 1C Lipoly que cuenta con las siguientes características técnicas:

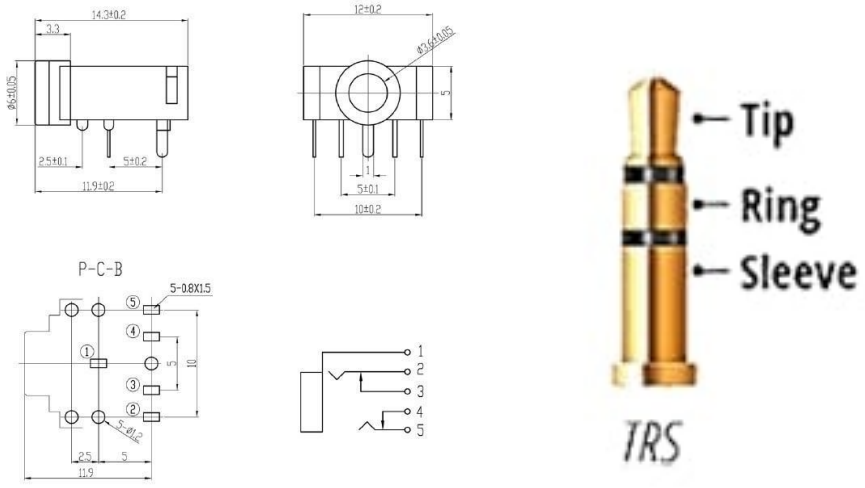
- Capacidad - 1200mAh
- Configuración - 1S / 3.7v / 1Cell
- Constante de descarga - 1C
- Peso - 20.5g
- Dimensiones - 48 x 33 x 6mm
- Conector carga/descarga - 2 pin JST-PH

Además de todos estos elementos más complejos usaremos componentes sencillos que se describen a continuación.

Conector Jack hembra

Conector Jack estéreo hembra de montaje en PCB, para dotar de salida de audio a nuestro prototipo.

- Tipo - Toma de 3,5 mm
- Pines - 5 pasadores
- Material - metal, plástico
- Tamaño: 1,4 x 1,2 x 0,5 cm / 0,6"" x 0,5"" x 0,2""(L*W*H)
- Tamaño del Pasador - 3 x 1mm// 0.1"" x 0.03""(L*W)
- Peso - 9g



Interruptor

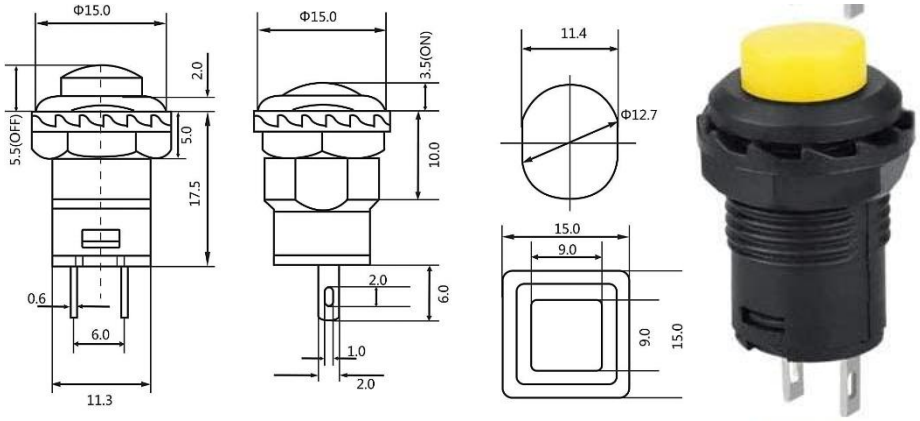
Interruptor de dos posiciones para encendido y apagado.

- Interruptor de palanca de 3 pines.
- Posición - 2 Posiciones (ON / OFF).
- Longitud terminal - 4 mm
- Tamaño total - 3.3 * 1.3 * 0.8cm.
- Vida mecánica - 1000000

Pulsadores momentáneos

Pulsadores que no mantienen su posición al ser accionados, es decir vuelven al estado de “abierto” una vez son pulsados.

- Orificio de montaje: aprox. 12mm
- Corriente y voltaje clasificado: 3A/125V AC; 1A/250V AC
- Intensidad dieléctrica: 1500VAC / 1min.
- Vida:> 10000 Ciclos
- Resistencia de contacto: <50mΩ
- Resistencia de aislamiento: > 100mΩ

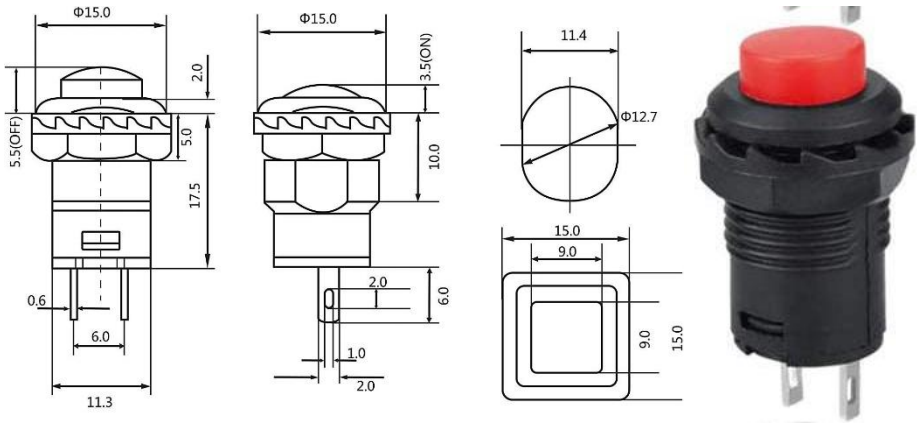


Pulsadores enclavados

El interruptor de enclavamiento es un interruptor que mantiene su estado después de ser activado.

Por lo tanto, cada vez que se active, cualquiera que sea el estado en el que permanezca el interruptor persistirá hasta que el interruptor vuelva a activarse.

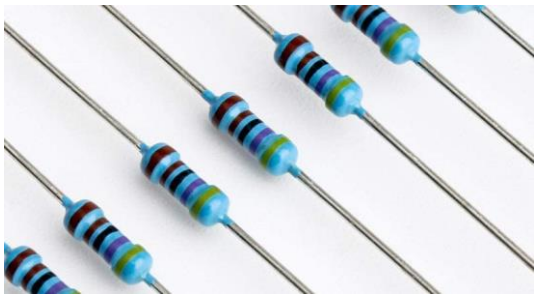
- Orificio de montaje: aprox. 12mm
- Corriente y voltaje clasificado: 3A/125V AC; 1A/250V AC
- Intensidad dieléctrica: 1500VAC / 1min.
- Vida:> 10000 Ciclos
- Resistencia de contacto: <50mΩ
- Resistencia de aislamiento: > 100mΩ



Resistencias

Varias resistencias genéricas de 1/2W.

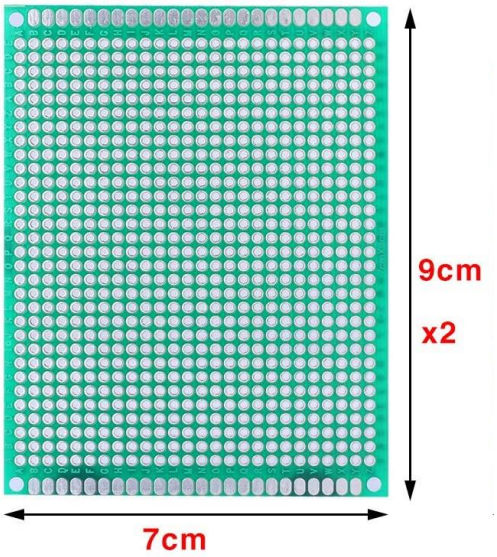
- 4k7Ω (x8)
- 1kΩ



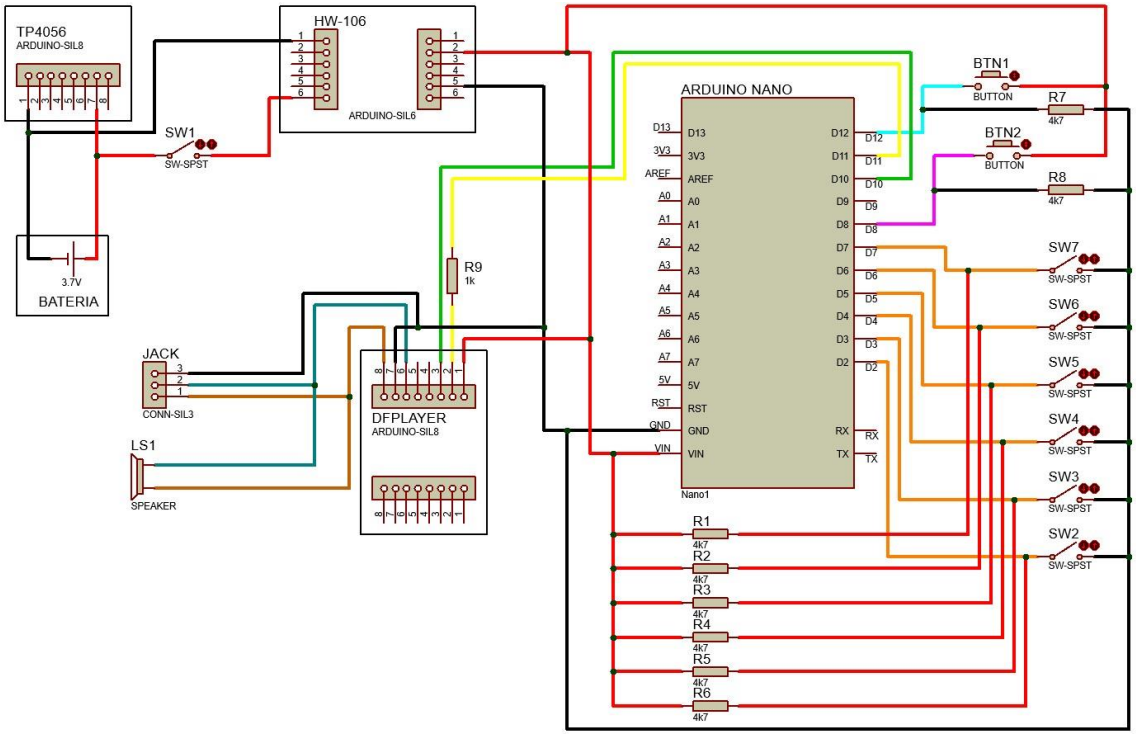
Placa micro perforada

Para realizar el montaje del prototipo de manera definitiva.

- Dimensión: 7x9 CM
- Espesor: alrededor de 1.6mm
- Material: fibra de vidrio FR4
- Paso del orificio: 2,54 mm/0,1 pulgadas
- Diámetro del orificio: 1 mm / 0.04 pulgadas



CIRCUITO



1. Alimentación:

La batería de 3.7V y 1200Ah se conecta con su borne negativo simultáneamente al borne negativo (BAT[-]) del módulo cargador TP4056 y al borne negativo del convertidor DC/DC HW-106. En el caso del borne positivo se conectará directamente al borne positivo (BAT[+]) del módulo cargador TP4056 y al borne positivo del convertidor DC/DC HW-106, sin embargo, en esta última línea introduciremos un interruptor (o switch) para poder cortar la corriente hacia el convertidor.

Este montaje en paralelo permite que la batería sea cargada mediante el módulo TP4056 y a la vez aporte una tensión a la entrada del convertidor (siempre que el interruptor este cerrado) al convertidor HW-106.

A la salida del convertidor DC/DC obtenemos 5V que alimenta el Arduino NANO (en el pin V_{in}) y al módulo DFPLAYER (en el pin VCC). Además, de esta línea obtenemos la referencia para los dos pulsadores (BTN1 y BTN2) que están en configuración de resistencia Pull-Down y los pulsadores con enclavamiento (SW2 a SW7) que están en configuración de resistencia Pull-Up.

La línea negativa a la salida de DC/DC será la tierra común a todo el circuito.

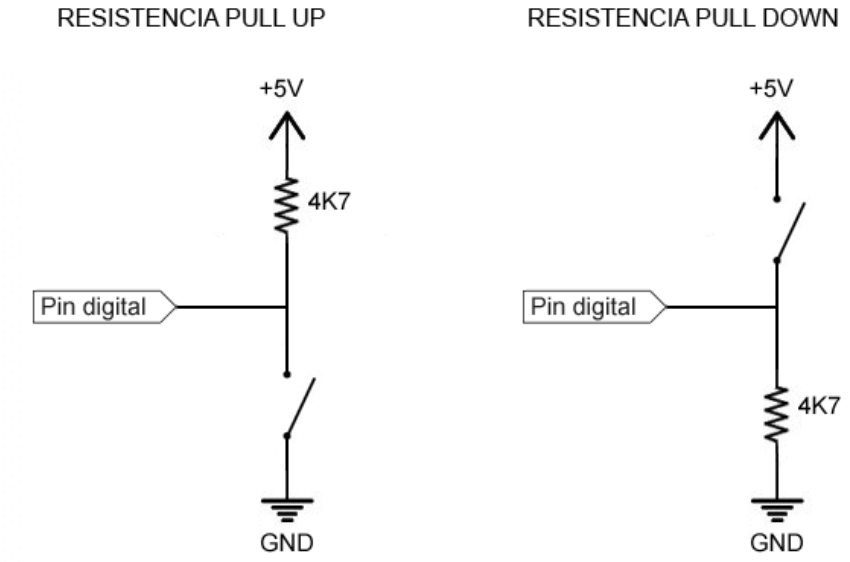
2. Arduino NANO:

Como hemos señalado, recibirá la alimentación desde el módulo DC/DC (de 5V) a través de los pines V_{in} y GND.

Los pines digitales del 2 al 7 serán pines de entrada donde se conectan los interruptores con enclavamiento en configuración Pull-Up mientras que los pulsadores sin enclavamiento se conectan al pin 8 y 12 (también como pines de entrada), ambos se encuentran en configuración Pull-Down. Para saber porque se ha elegido esta configuración en los dos casos hay que explicar la diferencia entre ambas:

Las resistencias de Pull-Down y Pull-Up se conectan entre el PIN digital y una de las tensiones de referencia (0V o 5V) y "fuerzan" (de ahí su nombre) el valor de la tensión a LOW o HIGH, respectivamente.

- *La resistencia de Pull-Up fuerza HIGH cuando el pulsador está abierto. Cuando está cerrado el PIN se pone a LOW, la intensidad que circula se ve limitada por esta resistencia.*
- *La resistencia de Pull-Down fuerza LOW cuando el pulsador está abierto. Cuando está cerrado el PIN se pone a HIGH, y la intensidad que circula se ve limitada por esta resistencia.*



Por lo tanto, los pulsadores en Pull-Up consumen corriente mientras están abiertos y los de Pull-Down, al contrario, solo cuando están cerrados.

Ya que los pulsadores sin enclavamiento están “normalmente abiertos” la resistencia en Pull-Down es la mejor opción para consumir menos. Sin embargo, los pulsadores con enclavamiento en nuestro diseño están cerrados en la situación de reposos del aparato por lo que colocarlos en configuración de Pull-Up es la mejor opción.

En los puertos 10 y 11 el Arduino se conecta al módulo DFPLAYER mediante comunicación puerto serie (Rx, Tx), está conexión se explica en el siguiente punto.

3. DFPlayer:

Como señalamos en el primer punto la alimentación (5V) vendrá directamente desde el módulo DC/DC a los pines VCC y GND (a cualquiera de los dos que tiene).

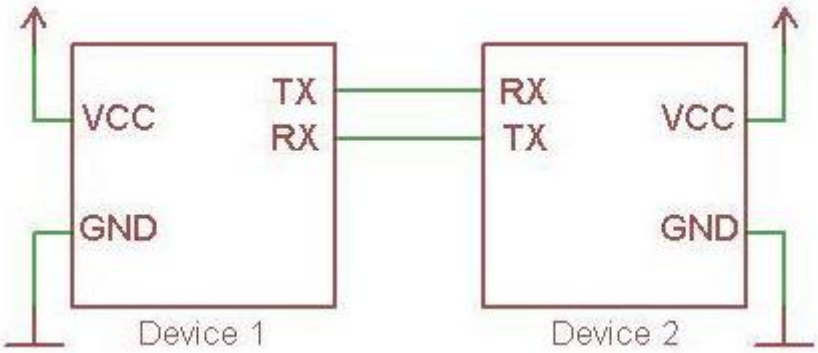
En los pines Spk_1 y Spk_2 se conectarán en paralelo el altavoz y el conector hembra Jack. En el Spk_1 una de las líneas del altavoz y la cabeza (Tip) del Jack hembra y al Spk_2 la otra línea del altavoz y el anillo (Ring) del Jack hembra. El cuerpo (Sleeve) del Jack hembra se conectará a tierra.

Para comprender la conexión entre el módulo DFPlayer y el Arduino NANO explicaremos de forma sencilla dicho protocolo.

Este protocolo se implementa en los receptores/transmisores asíncronos universales o UART (universally asynchronous receiver/transmitter) es una unidad que incorporan ciertos procesadores, encargada de realiza la conversión de los datos a una secuencia de bits y

transmitirlos o recibirlos a una velocidad determinada. Cuando un dispositivo se comunica usando el protocolo serial, su UART transmite en la línea "TX" y recibe datos por la línea "RX".

Cuando se conecta dos dispositivos de esta manera, un dispositivo envía la información usando su línea TX y el otro la recibe por su línea RX y viceversa.



Sólo un dispositivo puede tener control de una línea UART en un determinado momento. Si más de un dispositivo transmite a la línea RX de otro dispositivo, se produce un error llamado "contención de bus". Esto se traduce en que en vez de tener muchos dispositivos comunicándose con uno, éstos "competirán" por la línea de datos y no se podrá transmitir ni recibir información.

Por esta razón en el montaje se han configurado dos pines como "Rx" y "Tx" (10 y 11) para la comunicación en el Arduino NANO ya que los que vienen así configurados por defecto en la placa se comparten con el conector micro USB por lo que podrían entrar en conflicto al usarse a la vez conectado al PC para monitorizar su funcionamiento.

Así la conexión queda: del pin Tx del módulo mp3 al pin 10 del NANO que hemos configurado como Rx y del pin Rx del módulo mp3 al pin 11 del NANO configurado como Tx. En esta línea se ha añadido una resistencia de 1kΩ como recomienda el fabricante para evitar ruido en la señal de audio.

CODIGO

```
#include "Arduino.h"
#include "SoftwareSerial.h" //Biblioteca para el control de comunicación serial
#include "DFRobotDFPlayerMini.h" //Biblioteca para el control de módulo DFPLAYER
#include "stdlib.h"


SoftwareSerial mySoftwareSerial(10,11); // Pines por los que se va a realizar la comunicación serial (Rx, Tx) entre DFPLAYER y Arduino
DFRobotDFPlayerMini myDFPlayer;  //iniciación de módulo DFPLAYER
int CHECK, CHECK2, GAME, MODO; //Variables de tipo entero para "leer", "jugar" y cambiar de modo.
int PTEST = 12, PGAME = 8, P1 = 2, P2 = 3, P3 = 4, P4 = 5, P5 = 6, P6 = 7; //Nombres para los pines de la placa para facilitar el desarrollo y comprensión del código
int num, num_rand; //Variables de tipo entero para numerar las letras y para letra aleatoria
```

```
void printDetail(uint8_t type, int value); //Función de DFPLAYER que chequea el funcionamiento y devuelve los datos en "print"

void setup() {

  mySoftwareSerial.begin(9600); //Inicio de controlador de comunicación serial
  Serial.begin(9600); //Inicio de comunicación serial
  pinMode(P1, INPUT);
  pinMode(P2, INPUT);
  pinMode(P3, INPUT);
  pinMode(P4, INPUT);
  pinMode(P5, INPUT);
  pinMode(P6, INPUT);
  pinMode(PTEST, INPUT);
  pinMode(PGAME, INPUT);

  randomSeed(analogRead(A0)); //Lectura de puerto analógico libre para la obtención real de numero aleatorio

  if (!myDFPlayer.begin(mySoftwareSerial)) {   //Usa softwareSerial para comunicarse con el módulo mp3
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true){
      delay(0);
    }  }
  myDFPlayer.volume(30);  //Selección del valor de volumen entre 0 y 30
}

int abecedario() {
/* Con esta función damos un valor numérico entero a cada una de las combinaciones de los seis pulsadores que recrean el alfabeto Braille*/
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
  {num=1;}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
  {num=2;}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
  {num=3;}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
```

Configuramos cada puerto como entrada o salida

```
{num=4;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
{num=5;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
{num=6;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
{num=7;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
{num=8;}
if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
{num=9;}
if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
{num=10;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
{num=11;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
{num=12;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
{num=13;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
{num=14;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1))
{num=15;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
{num=16;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
{num=17;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
{num=18;}
if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
```

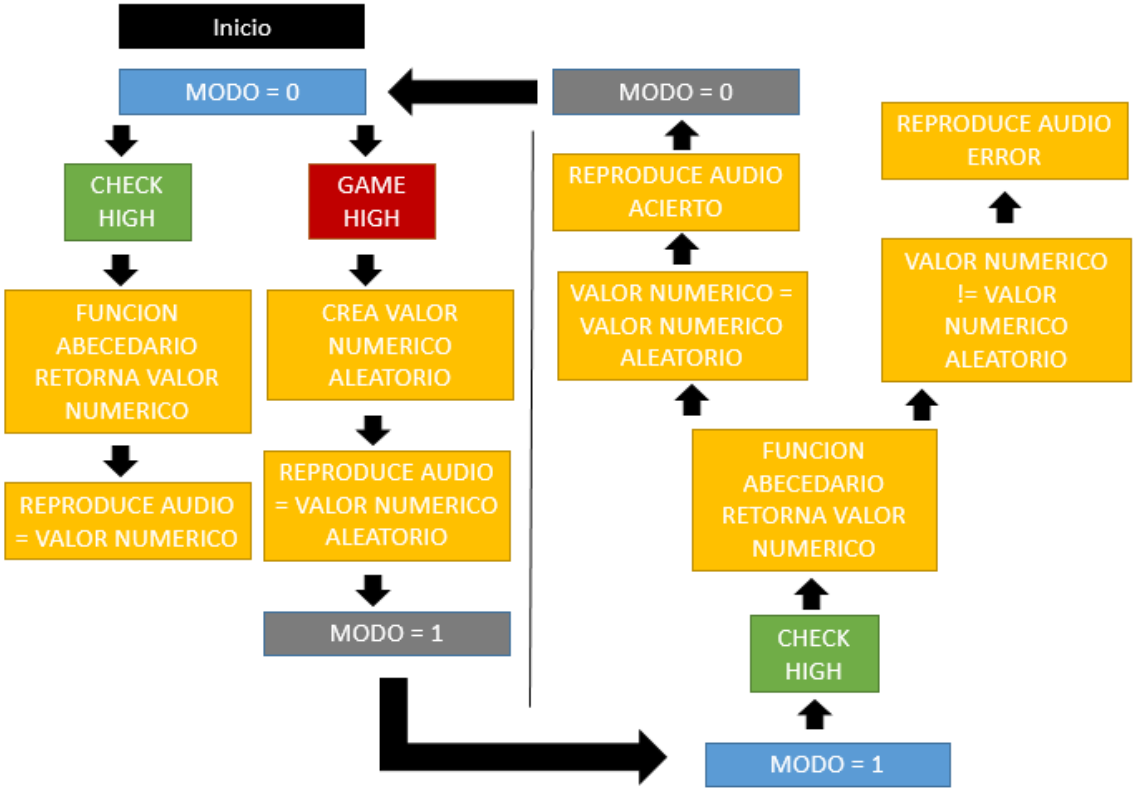
```
  {num=19;}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==0))
  {num=20;}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0))
  {num=21;}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1))
  {num=22;}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1))
  {num=23;}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1))
  {num=24;}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1))
  {num=25;}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1))
  {num=26;}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1))
  {num=27;}

  return num; //Tras el chequeo de la posición de los pulsadores devuelve el valor que corresponde
}

void loop() {

  num = abecedario(); //le damos el valor a la variable num que nos retorna la función abecedario
  CHECK = digitalRead(PTEST); //damos a la variable CHECK el valor de la lectura digital del pin PTEST (botón derecho)
  GAME = digitalRead(PGAME); //damos a la variable GAME el valor de la lectura digital del pin PGAME (botón izquierdo)


  if (CHECK == HIGH)  { //Si el botón derecho es pulsado entramos en este if
     delay (250); //delay para evitar el efecto rebote en la pulsación del botón
     myDFPlayer.play(num); //Se reproduce el audio del mismo número que coincide con la combinación de los pulsadores en la función abecedario
     Serial.println("leyendo");
     Serial.println(num); //se imprime el mismo número que coincide con la combinación de los pulsadores en la función abecedario
```

```
    delay(500);} //delay para evitar interferencias con la reproducción del audio

  if (GAME == HIGH){ //Si el botón izquierdo es pulsado entramos en este if
    delay (250); //delay para evitar el efecto rebote en la pulsación del botón
    num_rand = random(1,28); //obtenemos un valor aleatorio entre 1 y 27 (ambos incluidos) para la variable num_rand
    delay (150);
    myDFPlayer.play(num_rand); //Se reproduce el audio del mismo número que coincide con el numero aleatorio resultante
    Serial.println("pidiendo");
    Serial.println(num_rand); //Se imprime el audio del mismo número que coincide con el numero aleatorio resultante
    delay(1000); //delay para evitar problemas entre la reproducción de la letra aleatoria y el MODO
    MODO = 1;} //La variable MODO pasa a valer 1

  while (MODO == 1){ //Mientras la variable MODO vale 1 estaremos en este while

    CHECK2 = digitalRead(PTEST);  //damos a la variable CHECK2 el valor de la lectura digital del pin PTEST (botón derecho)
    num = abecedario(); //le damos el valor a la variable num que nos retorna la función abecedario

    if ((CHECK2 == HIGH) && (num_rand==num)){ //Si el botón derecho es pulsado y los valores de num_rand y num son iguales entramos en este if
      delay (250); //delay para evitar el efecto rebote en la pulsación del botón
      myDFPlayer.play(28); //se reproduce el audio de acierto
      Serial.println("BIEEEEEEEN"); //se imprime el mensaje de acierto
      delay (500); //delay para evitar problemas entre la reproducción del audio y el MODO
        MODO = 0;} //La variable MODO pasa a valer 0 y salimos del while

    if ((CHECK2 == HIGH) && (num_rand!=num)){ //Si el botón derecho es pulsado y los valores de num_rand y num son diferentes entramos en este if
      delay (250); //delay para evitar el efecto rebote en la pulsación del botón
      myDFPlayer.play(29); //se reproduce el audio de error
      Serial.println("OOOOOOOHH"); //se imprime el mensaje de error
      delay (500);} //La variable MODO sigue valiendo 1 asi que continuamos dentro del while
      }
}

void printDetail(uint8_t type, int value){ //Funcion de DFPLAYER que chequea el funcionamiento y devuelve los datos en "print"
  switch (type) {
    case TimeOut:
      Serial.println(F("Time Out!"));
```

```
  break;
case WrongStack:
  Serial.println(F("Stack Wrong!"));
  break;
case DFPlayerCardInserted:
  Serial.println(F("Card Inserted!"));
  break;
case DFPlayerCardRemoved:
  Serial.println(F("Card Removed!"));
  break;
case DFPlayerCardOnline:
  Serial.println(F("Card Online!"));
  break;
case DFPlayerUSBInserted:
  Serial.println("USB Inserted!");
  break;
case DFPlayerUSBRemoved:
  Serial.println("USB Removed!");
  break;
case DFPlayerPlayFinished:
  Serial.print(F("Number:"));
  Serial.print(value);
  Serial.println(F(" Play Finished!"));
  break;
case DFPlayerError:
  Serial.print(F("DFPlayerError:"));
  switch (value) {
    case Busy:
      Serial.println(F("Card not found"));
      break;
    case Sleeping:
      Serial.println(F("Sleeping"));
      break;
    case SerialWrongStack:
      Serial.println(F("Get Wrong Stack"));
      break;
    case CheckSumNotMatch:
      Serial.println(F("Check Sum Not Match"));
      break;
    case FileIndexOut:
      Serial.println(F("File Index Out of Bound"));
      break;
```

```
      case FileMismatch:
        Serial.println(F("Cannot Find File"));
        break;
      case Advertise:
        Serial.println(F("In Advertise"));
        break;
      default:
        break;      }
    break;
  default:
    break;
}}
```

DIAGRAMA DE BLOQUES



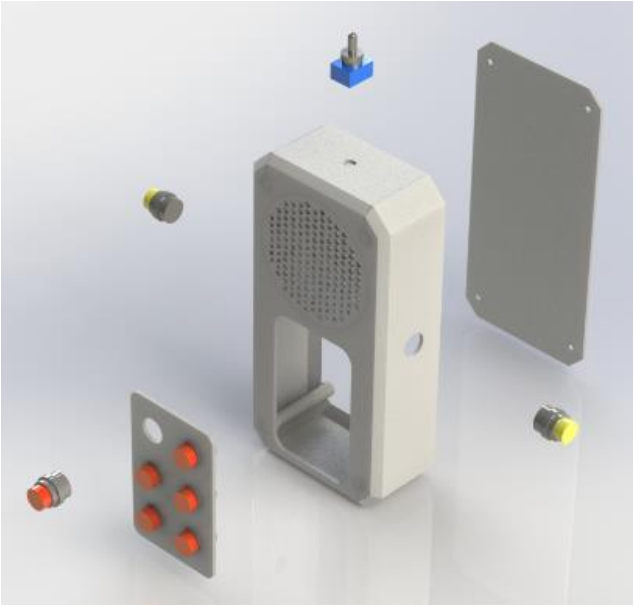
En el diagrama de bloques podemos ver como al iniciar la variable MODO tiene valor 0 lo que nos deja dos opciones. Con la pulsación del botón Check se efectuaría la lectura de la combinación de los pulsadores. Mientras que con la pulsación de Game el código crea un valor aleatorio y lo reproduce, a continuación, la variable MODO pasa a tener un valor de 1.

Con MODO igual a 1 pasamos al juego donde con el botón Check de nuevo se comprueba la combinación de los pulsadores, si esta es de un valor diferente se reproduce un audio de error y continua a la espera. Por el contrario, si el valor de la combinación de pulsadores es igual al aleatorio se reproduce el audio de acierto y la variable MODO pasa a valer 0 de nuevo, haciendo que el programa salga del juego al estado normal.

CARCASA

El concepto inicial para el diseño de la carcasa fue el de darle un aspecto de “cabeza de robot”, siguiendo esta idea las condiciones que llevaron al diseño final fueron:

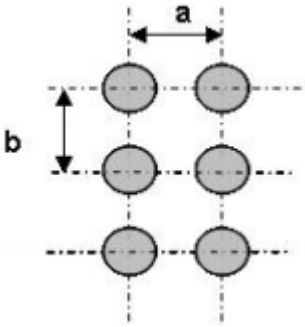
- Encontrar un tamaño manejable para las manos de un niño y donde poder alojar todos los componentes necesarios en el interior.
- Comodidad en el manejo y la sujeción.
- Compatibilidad con el sistema Braille.



El resultado a estas condiciones de diseño fue:

- Optamos por un tamaño similar al primer modelo de la GameBoy de Nintendo, colocando el altavoz en la posición de la pantalla. EL tamaño alargado de los pulsadores y el diámetro del altavoz impiden realizar un diseño más compacto.
- Para un agarre más cómodo realizamos un rebaje en todos los bordes para evitar esquinas de 90º que se clavaran durante el manejo. La botonera se elevó en un relieve sobre la carcasa para ayudar a encontrarla con facilidad por el usuario. Los pulsadores de lectura y juego se colocaron a la mitad de la carcasa para poder acceder a ellos sin tener que quitar las manos de la posición natural de uso.
- La medida oficial de separación entre los puntos (de centro a centro) es de 2’5mm de media así que lo multiplicamos por 10 para separar los pulsadores 2’5cm.

a = Distancia horizontal entre los centros de puntos contiguos de la misma
celda: de 2,4 a 2,75 mm.
b = Distancia vertical entre los centros de puntos contiguos de la misma
celda: de 2,4 a 2,75 mm.



“DOCUMENTO TÉCNICO B 1
DE LA
COMISIÓN BRAILLE ESPAÑOLA”

Resultado final del diseño v1.0. El diseño de la carcasa se ha realizado en colaboración con Víctor Rodríguez (Graduado en Ingeniería industrial) y Luis Rodríguez (Graduado en Ingeniería de diseño industrial).

PRESUPUESTO DE PROTOTIPO

COMPONENTES	PRECIO UNIDAD	PRECIO TOTAL	DECRIPCION
PLACA ARDUINO NANO (3)	4,67	4,67	ELEGOO UNO R3 Tarjeta Placa ATmega328P ATMEGA16U2 con Cable USB y Microcontrolador Compatible con Arduino IDE Proyectos Cumple con RoHS.
PULSADORES CON ENCLAVAMIENTO (18)	0,45 (X6)	2,7	BUYGOO 18 Piezas Interruptor de Botón Pestillo de Plástico Bricolaje 12 mm Mini Impermeable Botón Pulsador 6 Colores interruptores de Enclavamiento Encendido y Apagado.
PULSADORES MOMENTANEO (25)	0,36 (X2)	0,72	Runcci-Yun 25pcs Pulsador 12mm,Pulsadores Electricos Boton,Pulsador Mini,Pulsador Redondo,Interruptor Momentaneo 12v, Para Trompeta De Coche, Pc, Lámpara De Mesa, Timbre De Casa
MODULO CARGADOR DE BATERIA (5)	1,99	1,99	KKmoon Módulo de carga 5 unids 5 V 1A Micro USB 18650 Batería de Litio de Carga + Módulo de Cargador de Circuito de Protección módulo
MODULO ELEVADOR DC/DC (5)	1,86	1,86	TECNOIOT 5pcs 0.9V-5V DC-DC to 5V USB Voltage Converter Step Up Power Supply Module | 5pcs DC-DC 0.9V-5V a 5V USB Convertidor de Voltaje Booster Step up Módulo de alimentación
MODULO MP3 (3)	3,27	3,27	AZDelivery 3 x Mini MP3 Player DFPlayer Master Module para Arduino.
ALTAVOZ	5,99	5,99	HALJIA 1 W 8 Ohm 50 mm Dia magnético Tipo Redondo Metal Shell Altavoz Cuerno 50 MM Dia
INTERRUPTOR (12)	0,83	0,83	CESFONJER 12 Piezas Mini Interruptor de Palanca, AC 125V 6A ON-OFF-ON 3 Pines 3 Posiciones SPDT Interruptor Rocker Switch de Azul
CONECTOR JACK HEMBRA (10)	0,55	0,55	Sourcingmap® 10 x PCB de Montaje en Panel 5-Pin Hembra estéreo de 3,5 mm Conectores Hembra Jacks
BATERIA	4,05	4,05	Turnigy 1200mAh 1S 1C Lipoly w/ 2 Pin JST-PH Connector
RESISTENCIAS	0,15 (x9)	1,35	Resistencia generica de 1/2W
PLACA MICROPERFORADA 7X9CM (32)	3	3	ELEGOO 32 Piezas Doble Cara Junta de PCB Doble Cara Tarjeta Placa Prototipo Kit para Soldadura DIY con 5 Tamaños Compatible con los kits Arduino
	TOTAL	30,98	

A este coste habría que sumar el valor de la carcasa impresa en 3D, este cálculo es complejo ya que depende del tipo de impresora utilizada y dentro de esta, el material que se use (composición y calidad). En este caso se ha empleado una impresora de filamento con PLA 1.75 blanco que tiene un coste de 19.99€/kg para una pieza de 150g el coste aproximado sería de 2.99€.

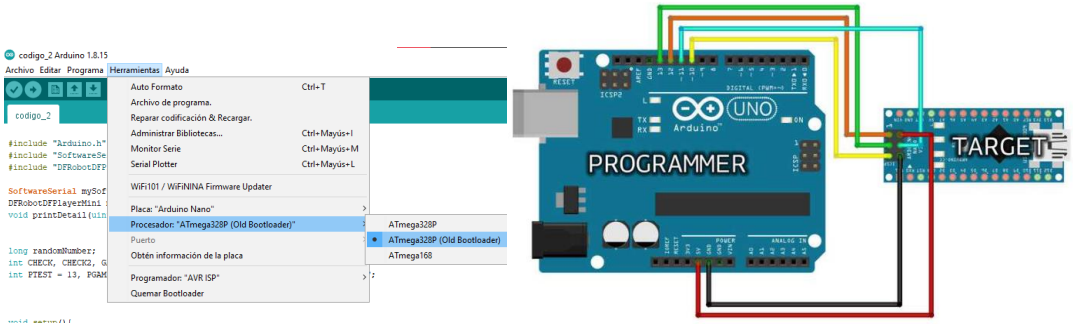
CONCLUSIONES

El trabajo de desarrollo de código ha sido relativamente sencillo, los principales inconvenientes encontrados han sido hallar la manera de adjudicar un valor numérico a las diferentes combinaciones de los pulsadores que recrean el alfabeto Braille y el funcionamiento del módulo reproductor DFPlayer.

En lo que se refiere al Hardware el gran escollo ha sido la programación del Arduino NANO debido a una incompatibilidad del IDE actual de Arduino con algunos chips ATmega328P que montan las versiones baratas del modelo NANO.

Después de buscar información y visionar varios tutoriales para reparar este error se realizaron las siguientes acciones:

- Probar en diferentes ordenadores.
- Probar con diferentes IDEs (portable, instalado, aplicación W10…).
- Instalar nuevos Drivers de NANO.
- Probar en el IDE con la opción para Bootloader antiguo.
- Conectar el Arduino NANO a una placa de Arduino UNO para usándola como ISP actualizar el Bootloader desde esta.



Ninguna de estas soluciones dio resultado por lo que finalmente hubo que optar por la sustitución del Arduino NANO por otro diferente al que sí se le pudo cargar el código correctamente.

AMPLIACIONES

Las ideas de mejoras, para este prototipo, surgidas durante el desarrollo del proyecto y las reuniones con el profesorado son las siguientes:

Mejoras posibles a nivel software:

- No salir del modo juego al acertar, es decir que siga pidiendo letras hasta que el usuario decida salir.
- Variantes al juego de pedir letra buscar, varios modos de juego.
- Añadir conmutadores que permitan cambiar entre modos para incluir números y símbolos en las combinaciones.

Mejoras posibles del hardware:

- Añadir potenciómetro para el control del volumen por el usuario.
- Añadir etapa de potencia a la salida del DFPlayer.

- Diseñar PCB con zócalos y conectores para una rápida instalación y mantenimiento.

Mejoras posibles del diseño:

- Subir los pulsadores de juego y lectura para una ergonomía y manejo mejores.
- Usar conector Jack hembra de montaje en chasis en lugar de PCB para una mejor instalación, imagen en Anexo III.
- Usar pulsadores más cortos para poder mejorar el tamaño del modelo.
- Acceso desde el exterior del puerto microUSB del Arduino NANO.

Una vez completado este proyecto, la mejor vía de investigación para la siguiente versión seria cambiar el microcontrolador por un microprocesador, tipo Raspberry Pi, lo que abriría un gran abanico de opciones y mayor versatilidad y conectividad. Entre otras, por ejemplo:

- Lector de textos
- Aplicación móvil que pudiera conectarse al dispositivo para seguimiento del profesorado.

Durante el proceso, pensando en las edades de los niños que podrían utilizar nuestro prototipo, surgió la idea de crear un modelo de un tamaño superior (y más resistente) para que niños más pequeños den los primeros pasos y comiencen a familiarizarse con el sistema Braille.

WEBGRAFÍA

https://radiomitre.cienradios.com/braillin-la-historia-del-muneco-que-cambio-la-manera-de-ensenar-braille/

https://www.tecnobility.com/es/noticia/juguetes-para-acercar-poco-poco-el-braille-ni%C3%B1os-con-discapacidad-visual

https://purodiseno.lat/innovacion/cinco-amigas-desarrollaron-un-teclado-y-una-impresora-que-permite-escribir-e-imprimir-en-braille/

https://www.hwlibre.com/arduino-nano/?utm_source=dlvr.it&utm_medium=twitter

https://es.wikipedia.org/wiki/Microcontrolador

https://www.electroallweb.com/index.php/2020/07/22/modulo-dfplayer-mini-reproductor-mp3-tutorial-completo/

https://www.luisllamas.es/arduino-mp3-dfplayer-mini/

https://uelectronics.com/producto/mini-dfplayer-mp3-para-arduino/

https://es.wikipedia.org/wiki/Convertidor_elevador

https://es.wikipedia.org/wiki/Altavoz

https://www.digikey.com/es/products/detail/cui-devices/GF0771M/304454

https://hobbyking.com/es_es/turnigy-1200mah-1s-1c-lipoly-w-2-pin-jst-ph-connector.html?countrycode=ES&gclid=CjwKCAjw_JuGBhBkEiwA1xmbRXsYkZaeffx5Y0n1kZNBIz06IAZo1XlHecA81eLzKByZWbSa89Pn3RoC9AMQAvD_BwE&gclsrc=aw.ds

https://www.luisllamas.es/leer-un-pulsador-con-arduino/

file:///C:/Users/Jimbo/Desktop/B1.%20PARAMETROS%20DIMENSIONALES%20DEL%20BRAILLE%20V1.pdf

https://www.luisllamas.es/arduino-puerto-serie/

https://cursos.mcielectronics.cl/2019/06/18/senales-tx-y-rx/

https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299

https://forum.arduino.cc/t/problemas-al-subir-programa-en-arduino-nano/323461

https://www.youtube.com/watch?v=LtDz15h0c9w

https://www.youtube.com/watch?v=eU0nIb1EGbo

https://www.arduino.cc/en/Tutorial/BuiltInExamples/ArduinoISP

https://www.youtube.com/watch?v=LIWn59qNbRk&t=8s

ANEXOS

ANEXO I

```
long randomNumber;
int CHECK, CHECK2, GAME, MODO;
int PTEST = 13, PGAME = 12, LED = 11, P1 = 2, P2 = 3, P3 = 4, P4 = 5, P5 = 6, P6 = 7, ZMB
= 10;
void setup()
{
  Serial.begin(9600);
  pinMode(P1, INPUT);
  pinMode(P2, INPUT);
  pinMode(P3, INPUT);
  pinMode(P4, INPUT);
  pinMode(P5, INPUT);
  pinMode(P6, INPUT);
  pinMode(PTEST, INPUT);
  pinMode(PGAME, INPUT);
  pinMode(LED, OUTPUT);

  randomSeed(analogRead(A0));
}

void acierto(){
  tone(ZMB,1319,125);
```

```
  delay(130);
  tone(ZMB,1568,125);
  delay(130);
  tone(ZMB,2637,125);
  delay(130);
  tone(ZMB,2093,125);
  delay(130);
  tone(ZMB,2349,125);
  delay(130);
  tone(ZMB,3136,125);
  delay(125);
  noTone(ZMB);
  delay (500);}

void error(){
  tone(ZMB,165,125);
  delay(130);
  tone(ZMB,196,125);
  delay(130);
  tone(ZMB,98,125);
  delay(130);
  tone(ZMB,73,125);
  delay(130);
  tone(ZMB,65,125);
  delay(130);
  tone(ZMB,82,125);
  delay(125);
  noTone(ZMB); }

void loop() {

CHECK = digitalRead(PTEST);
GAME = digitalRead(PGAME);
digitalWrite(LED, LOW);
```

```
if (CHECK == HIGH){
   delay (150);
  if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1)) {
          Serial.println("a");}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
                    Serial.println("b");}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
                    Serial.println("c");}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
                    Serial.println("d");}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
                    Serial.println("e");}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
                    Serial.println("f");}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
                    Serial.println("g");}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
                    Serial.println("h");}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
                    Serial.println("i");}
  if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
                    Serial.println("j");}
  if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
                    Serial.println("k");}
```

```
 if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){

                Serial.println("l");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){

                Serial.println("m");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){

                Serial.println("n");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){

                Serial.println("o");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){

                Serial.println("p");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){

                Serial.println("q");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){

                Serial.println("r");}

 if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){

                Serial.println("s");}

 if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){

                Serial.println("t");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0)){

                Serial.println("u");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0)){

                Serial.println("v");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==0)){

                Serial.println("x");}

 if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==0)){

                Serial.println("y");}
```

```
 if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==0)){
                    Serial.println("z");}
 }

 if (GAME == HIGH){
  delay (150);
  randomNumber = random(1,28);
  delay (150);
  if (randomNumber == 1){
  Serial.println("a");}
  if (randomNumber == 2){
  Serial.println("b");}
  if (randomNumber == 3){
  Serial.println("c");}
  if (randomNumber == 4){
  Serial.println("d");}
  if (randomNumber == 5){
  Serial.println("e");}
  if (randomNumber == 6){
  Serial.println("f");}
  if (randomNumber == 7){
  Serial.println("g");}
  if (randomNumber == 8){
  Serial.println("h");}
  if (randomNumber == 9){
  Serial.println("i");}
  if (randomNumber == 10){
  Serial.println("j");}
  if (randomNumber == 11){
  Serial.println("k");}
  if (randomNumber == 12){
  Serial.println("l");}
  if (randomNumber == 13){
```

```
  Serial.println("m");}
  if (randomNumber == 14){
  Serial.println("n");}
  if (randomNumber == 15){
  Serial.println("ñ");}
  if (randomNumber == 16){
  Serial.println("o");}
  if (randomNumber == 17){
  Serial.println("p");}
  if (randomNumber == 18){
  Serial.println("q");}
  if (randomNumber == 19){
  Serial.println("r");}
  if (randomNumber == 20){
  Serial.println("s");}
  if (randomNumber == 21){
  Serial.println("t");}
  if (randomNumber == 22){
  Serial.println("u");}
  if (randomNumber == 23){
  Serial.println("v");}
  if (randomNumber == 24){
  Serial.println("w");}
  if (randomNumber == 25){
  Serial.println("x");}
  if (randomNumber == 26){
  Serial.println("y");}
  if (randomNumber == 27){
  Serial.println("z");}
  delay (150);
  MODO = 1;
        if (MODO == 2){
                MODO = 0;}
}
```

```
 while (MODO == 1){
CHECK2 = digitalRead(PTEST);


if (CHECK2 == HIGH){
  delay (250);
    if (randomNumber==1){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) &&
(digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1) ){
        acierto();
        delay (500);
        MODO = 0;}
       else{
           error ();}}
    if (randomNumber==2){
      if((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) &&
(digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1) ){
          acierto();
          delay (500);
          MODO = 0;}
       else{
           error ();}}
    if (randomNumber==3){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) &&
(digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
          acierto();
          delay (500);
          MODO = 0;}
       else{
          error ();}}
   if (randomNumber==4){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) &&
(digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
          acierto();
          delay (500);
```

```
        MODO = 0;}
      else{
        error ();}}
  if (randomNumber==5){
    if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
    else{
        error ();}}
  if (randomNumber==6){
    if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
    else{
        error ();}}
  if (randomNumber==7){
    if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
    else{
        error ();}}
  if (randomNumber==8){
    if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
    else{
        error ();}}
```

```
    if (randomNumber==9){
      if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
          acierto();
          delay (500);
          MODO = 0;}
      else{
            error ();}}
    if (randomNumber==10){
      if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
          acierto();
          delay (500);
          MODO = 0;}
      else{
            error ();}}
    if (randomNumber==11){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
          acierto();
          delay (500);
          MODO = 0;}
      else{
            error ();}}
    if (randomNumber==12){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
          acierto();
          delay (500);
          MODO = 0;}
      else{
            error ();}}
    if (randomNumber==13){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
          acierto();
```

```
        delay (500);
        MODO = 0;}
      else{
          error ();}}
    if (randomNumber==14){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
      else{
          error ();}}
    if (randomNumber==15){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==0)){
        acierto();
        delay (500);
        MODO = 0;}
      else{
          error ();}}
    if (randomNumber==16){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
      else{
          error ();}}
    if (randomNumber==17){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
      else{
```

```
        error ();}}
    if (randomNumber==18){
     if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
     else{
        error ();}}
    if (randomNumber==19){
     if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
     else{
        error ();}}
    if (randomNumber==20){
     if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
     else{
        error ();}}
    if (randomNumber==21){
     if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==1)){
        acierto();
        delay (500);
        MODO = 0;}
     else{
        error ();}}
    if (randomNumber==22){
     if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0)){
```

```
        acierto();
        delay (500);
        MODO = 0;}
      else{
          error ();}}
    if (randomNumber==23){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==0) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==1) && (digitalRead(P6)==0)){
        acierto();
        delay (500);
        MODO = 0;}
      else{
          error ();}}
    if (randomNumber==24){
      if ((digitalRead(P1)==1) && (digitalRead(P2)==0) && (digitalRead(P3)==1) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==0)){
        acierto();
        delay (500);
        MODO = 0;}
      else{
          error ();}}
    if (randomNumber==25){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==1) && (digitalRead(P6)==0)){
        acierto();
        delay (500);
        MODO = 0;}
      else{
          error ();}}
    if (randomNumber==26){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==0) && (digitalRead(P5)==0) && (digitalRead(P6)==0)){
        acierto();
        delay (500);
        MODO = 0;}
```

```
      else{
          error ();}}
    if(randomNumber==27){
      if ((digitalRead(P1)==0) && (digitalRead(P2)==1) && (digitalRead(P3)==0) && (digitalRead(P4)==1) && (digitalRead(P5)==0) && (digitalRead(P6)==0)){
          acierto();
          delay (500);
          MODO = 0;}
      else{
          error ();}}
  }}
}
```

ANEXO II

Conector Jack hembra de montaje en chasis