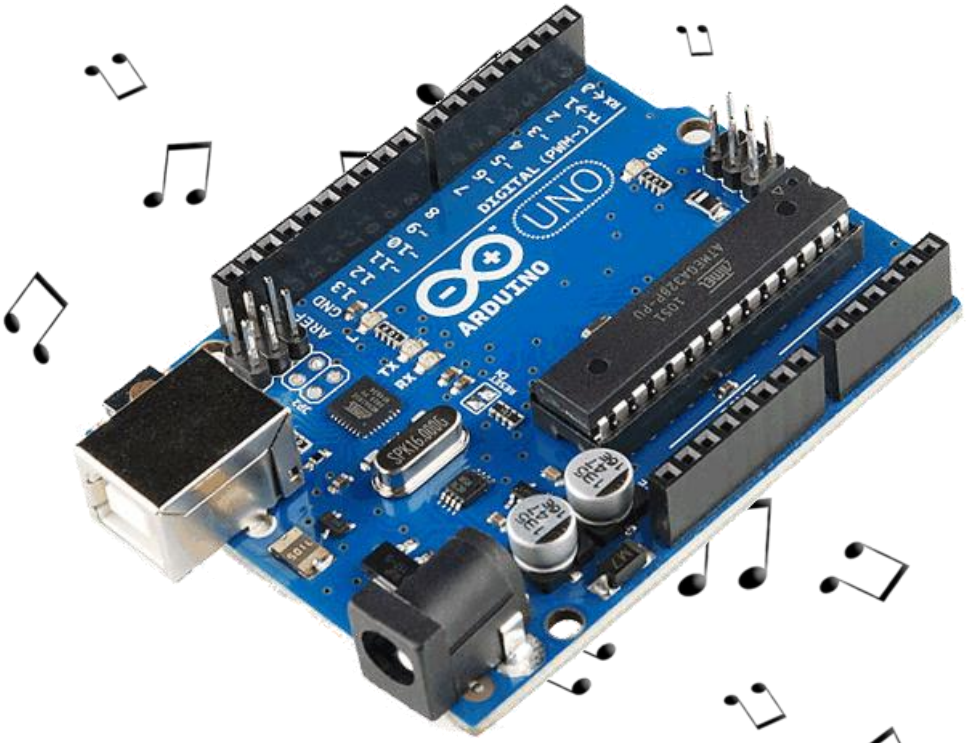
Organización y arquitectura de computadoras

Comisión 6

Arduino Music



Integrantes:

- Corsiglia Gonzalo
- Gómez Lucas

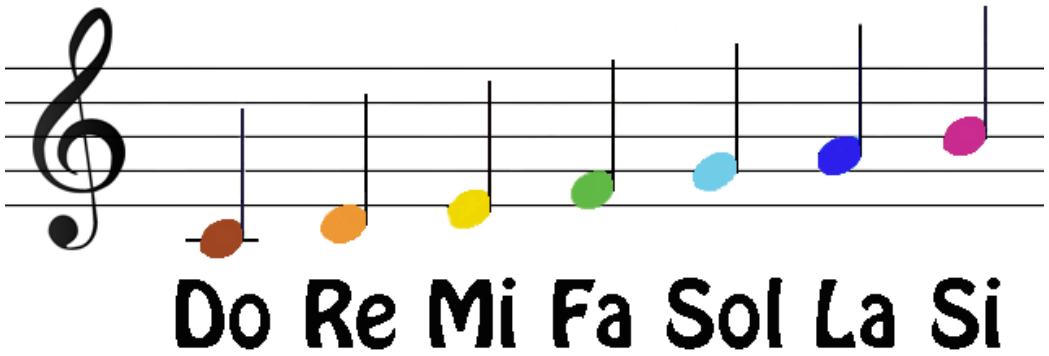
Introducción

“Al mal tiempo, buena música” dice el refrán, y a partir de tal cita es que damos comienzo a este recorrido.

La música forma parte de nuestras vidas, es ese faro que nos guía cuando nos encontramos perdidos o aquel impulso que necesitamos para tener un mejor día. La música impartida desde una edad temprana favorece el desarrollo intelectual beneficiando la alfabetización, la mejora de la concentración y la atención, potencia la memoria, estimula y desarrolla el habla, y ayuda a desarrollar la expresión y comprensión oral.

Haciendo hincapié en lo anteriormente mencionado, desde nuestro humilde grupo proponemos crear una herramienta didáctica y divertida, ambientada en la música y destinada al uso pedagógico en instituciones educativas.

A grandes rasgos, para convertir tal idea en una realidad, necesitaremos un teclado que cuente con un parlante para reproducir notas musicales.



De esta forma los finalmente usuarios de la herramienta, podrán jugar y divertirse creando combinaciones de sonidos espectaculares. El sistema también tendrá algunas canciones precargadas para agregar otro sabor a la experiencia.

Aunque esto ya pinta genial, nos parece que la implementación de una pantalla en la cual se pueda visualizar la nota musical que se esta reproduciendo es indispensable para un aprendizaje satisfactorio.

Y, por último, aunque no menos importante, agregaremos una luz led roja… ¿Por qué? No hay porque, a los niños les encantan los colores.

Sigamos…

Entrando más en detalle, la aplicación será en lenguaje C en base a las Librerías disponibles de la Arduino uno sobre el Microprocesador ATMEGA del Microcontrolador ATMEGA328p.

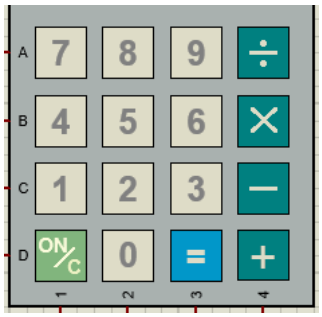
Se dispondrá con variados periféricos con sus respectivas librerías para la codificación en C, de los cuales utilizaremos aquellos que se amolden a la idea planteada anteriormente.

Una vez dado un pantallazo sobre nuestra propuesta, ya va siendo hora de hacerla realidad.

Descripción

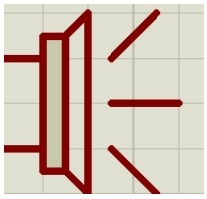
Los periféricos que utilizaremos serán:

Teclado

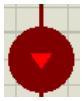


LCD

Parlante



Led Rojo



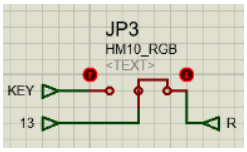
Para poder utilizar el teclado y el LCD necesitamos utilizar las siguientes librerías:

Keypad.h para el teclado.

LiquidCrystal.h para el LCD.

El parlante y el Led rojo, no necesitan librerías para su funcionamiento.

No obstante, el led necesita del siguiente jumper



En un principio debemos establecer los pines a los cuales están conectados los periféricos:

```
//LCD
LiquidCrystal lcd(1, 0, A1, A2, 10, 9);

//Parlante
int tonePin = 3;
//LED
int rojo = 13;
```

```
//Teclado
const byte ROWS = 4;
const byte COLS = 4;
```
Filas y columnas

```
char keys[ROWS][COLS] = {
  {'7', '8', '9', 'A'},
  {'4', '5', '6', 'B'},
  {'1', '2', '3', 'C'},
  {'E', '0', 'F', 'D'}
};
byte rowPins[ROWS] = {12, 11, 8, 7}; //Filas(pines del 7,8 y 12, 13)
byte colPins[COLS] = {6, 5, 4, 2}; //Columnas (pines del 2, 4 al 6)
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```
Formato

Constructor

Todo esto es instanciado fuera del setup y del loop de tal forma que estos puedan acceder globalmente a los objetos.

Luego dentro del `void setup()` ponemos el codigo de inicializacion de configuracion que tan solo se ejecutara una vez. En nuestro caso, inicializamos el lcd con su tamaño respectivo (metodo begin). Ademas, configuramos los pines del led y del parlante como una salida. Por ultimo, codeamos una frase que aparecera en el lcd, para brindar la estetica que la aplicación necesita.

```
void setup() {
  pinMode(tonePin, OUTPUT);
  analogWrite(tonePin, 0);
  pinMode(rojo, OUTPUT);
  lcd.begin(16, 2);
  lcd.print("Notas musicales");


}
```

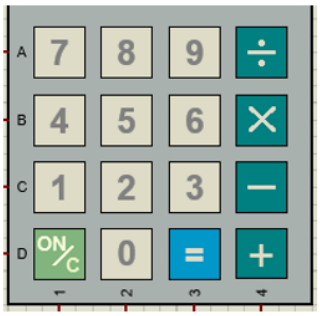
Ahora, vayamos al `void loop()`, dentro de este se encuentra el codigo que se ejecutara repetidamente. Necesitamos que al tocar una tecla suene una nota musical y al mismo tiempo se prenda la luz led y aparezcan en el lcd la nota musical que se reprodujo.

Para esto, creamos una variable que recibira el carácter que se ha presionado en el teclado

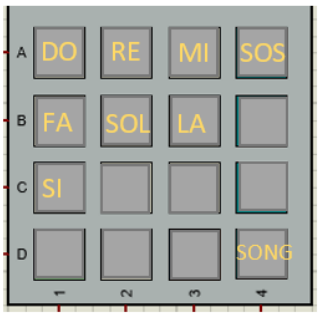
```
char key = 'x';
key = keypad.getKey();
```

Luego, utilizamos un switch (key), instanciando variados caminos que el programa tomara según la tecla que se presione del teclado.

Teclado 4x4 de Arduino



Arduino Music



Con este gráfico, queremos demostrar las equivalencias que hay entre el teclado que visualizamos y la funcionalidad de la herramienta.

Pero... Cuando el switch desemboca en alguno de los caminos que llevan a una nota musical ¿Cómo es que se lleva a cabo la reproducción de estos sonidos?, Fácil, la respuesta se encuentra en un método

Este metodo recibe una nota como string, y una frecuencia como entero. Con estos datos pasados por parametro, reproduce el sonido (metodo tone), imprime en pantalla la nota para asi saber que esta sonando.

Por ultimo, prende y apaga el led rojo, no preguntes el porque.

```
void sonido(String nota, int frecuency)
{
  tone(tonePin, frecuency, 200);
  delay(300);
  lcd.clear();
  lcd.setCursor(7, 0);
  lcd.print(nota);
  analogWrite(rojo, 255);
  delay(200);
  analogWrite(rojo,0);

}
```

Como te abras dado cuenta, en el grafico aparece una funcionalidad “SOS”, esta sirve para reproducir notas musicales sostenidas. Al tocar tal tecla, se activará el booleano

sostenido, dando a entender al programa que entre por el camino de notas sostenidas y devolviendo su sonido. Luego de la reproducción, el booleano vuelve a false.

Para este entonces, ya te disté cuenta de cuál es la magia de este dispositivo. Pero aún hay más.

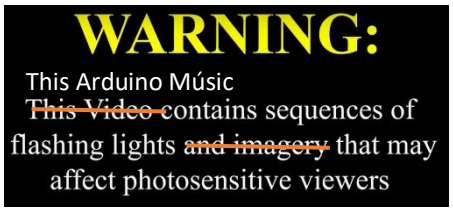
Como dijimos en la introducción, también contaremos con la reproducción de algunas canciones.

Entonces si presionamos la tecla con la funcionalidad “song”, comenzara a sonar una canción, “Mi corazón encantado de Dragon Ball GT”. ¿Podríamos haber escogido otra canción? Si. ¿Lo hicimos? No.

La canción se reproduce a través de la implementación de variados métodos, Playtone y Playnote. Estos métodos se encargan de darle el sonido y el tempo que la canción necesita para su desarrollo y reproducción satisfactoria.

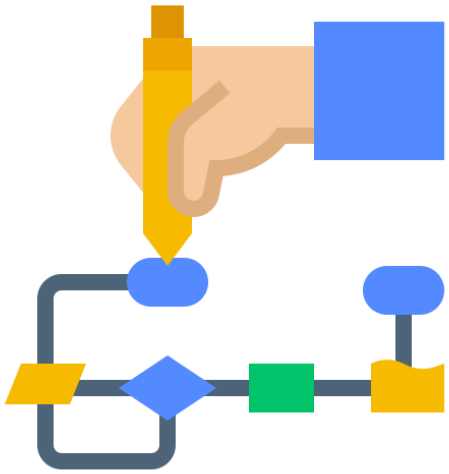
Y en esta oportunidad no va a ser la excepcion, tambien aparece a luz led roja.

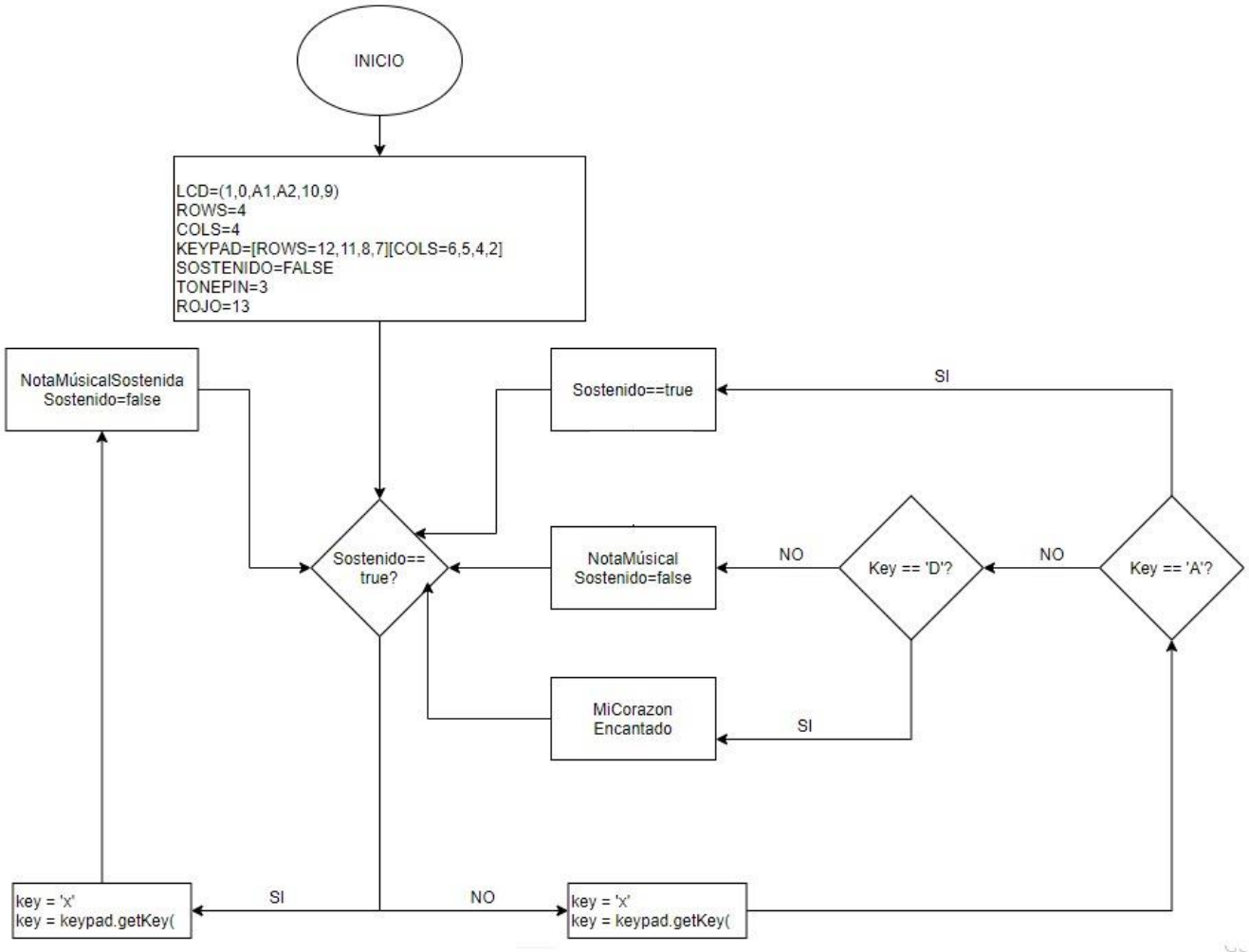
Para agregar otra funcionalidad curiosa, el metodo titilar nos deleitara con unos juegos de luces rojas letales.




```
void titilar(int retraso)
{
  analogWrite(rojo,255);
  delay(retraso);
  analogWrite(rojo,0);
  }
```

Y antes de pasar a los Resultados, queremos mostrarte el diagrama de flujo correspondiente a la aplicación, para terminar de cerrar la idea





Resultados

Los resultados fueron los previstos. Buscábamos una aplicación formativa, alegre y encantadora, capaz de cautivar los sentidos de los infantes en busca de fomentar el conocimiento sobre la música. Y creemos que lo hemos logrado. ¿Vos qué pensas?

Durante el recorrido, hemos realizado muchas pruebas en busca de la mayor calidad y afinación posible para ofrecer una aplicación a la altura de las expectativas planteadas.

En cuanto a las problemáticas encontradas, la implementación de la mecánica de las notas sostenidas conllevo una reflexión en busca de darle fin a tal cuestión. Por otro lado, contamos con una placa base (Arduino Uno) que no soporta tanta carga, por lo cual nos vimos obligados a recortar la cantidad de funcionalidades planteadas.

Conclusiones

Ya desde un punto de vista más personal, con mi compañero consideramos que este trabajo fue una experiencia positiva y enriquecedora.

La forma en que los distintos periféricos se conectan y a través del código se les puede brindar una funcionalidad a algo tangible, aunque en este caso simulado con Proteus… es una locura.

Aunque nos hubiera gustado agregar más canciones, y quizás entrar en detalle un poco mas con la aplicación, implementando métodos mas personalizados y rápidos, creemos haber cumplido con las expectativas. Y por ello estamos contentos.

Gracias.

Bibliografía

. https://www.arduino.cc/reference/en/language/functions/advanced-io/notone/

. https://www.youtube.com/watch?v=MISh3rSZcBg&ab_channel=SebastianCuenca

. https://github.com/ElectroCoderEC/proyectosMusicales

. https://www.youtube.com/watch?v=X4KBVEHvqfM&ab_channel=Rom%C3%A1nBond