

# LIGHTSABER AMADOR

Docente: Rui Escadas



Trabalho elaborado por:

Afonso Lopes (107697)

Inês Monteiro (107540)

Afonso Vaz (107838)

Marta Moniz (107209)

UA - Universidade de Aveiro

2022/2023

## Objetivos do Projeto

Este trabalho foi realizado no âmbito do módulo de Competências Transferíveis II, Microcontroladores e Interação Com Sensores e Atuadores. O principal objetivo reside na replicação das diversas funcionalidades de um *lightsaber*, inspirado da saga de *Star Wars*, através do uso de uma placa de desenvolvimento STM32F411RE e diversos sensores e atuadores fornecidos pelo docente do módulo. Após a implementação do algoritmo, o próximo passo consiste na construção tanto da lâmina como do punho do *lightsaber*, de forma a incorporar todos os dispositivos utilizados, e “sentir a sensação de ser um com a força”.

Este relatório tem como objetivo retratar o funcionamento do nosso projeto, tal como expor todo o processo, bem como todas as dificuldades, ao longo do desenvolvimento do nosso *lightsaber* amador.

## Desenvolvimento do Projeto

- PARTE I - Acelerómetro

Para o desenvolvimento deste dispositivo utilizámos um acelerómetro ADXL345 para medir a aceleração instantânea dos seus três eixos, que serve para o cálculo do módulo da aceleração ao longo do eixo do *lightsaber*. Este foi usado para determinar se o dispositivo se encontra estacionário, em movimento ou perante uma colisão, onde aciona diferentes mecanismos. Para o caso de estar estacionário existe a emissão de um som em *loop* (“*idle sound*”), para caso de estar em movimento, e dentro de um certo intervalo para o módulo de aceleração, parava o “*idle sound*” e emitia outro som (“*swing*”) e, para o último caso em que ocorresse uma colisão, emitiria um *flash* de luz branca e um terceiro som (“*hit*”).

- PARTE II - LED's

Para além disso, queríamos que o *lightsaber* fosse capaz de reproduzir as cinco cores originais apresentadas ao longo dos nove filmes da saga (verde, azul, roxo, amarelo e vermelho), bem como simular o efeito de *flash* aquando de uma colisão e o efeito de degradê e em escada ao ligar/desligar o dispositivo.

Para isto, utilizou-se uma fita LED endereçáveis, WS2812B, que resultou no nosso primeiro grande problema. Para os leds serem programáveis era necessário o uso da biblioteca *FastLED*, que por sua vez não era compatível com a placa STM32F411RE. Para contornar esta dificuldade optou-se pelo uso da placa Arduino UNO R3, uma vez que nos apercebermos que tínhamos sucesso quando compilávamos o código para uma placa de Arduino, o que não acontecia no caso de compilarmos para a placa fornecida pelo docente.

Nos últimos dias de desenvolvimento do projeto, a componente verde do LED número 12 ficou danificada, tendo nós decidido mantê-lo completamente desligado para todas as cores que requeriam da componente verde. Para além disso, verificámos que a fita tinha o seu código RGB trocado, pelo que a nossa implementação do código foi em função das cores na ordem GRB.

### • PARTE III - Coluna

Por fim, e para a reprodução de todos os efeitos sonoros, começámos por utilizar o buzzer que nos foi fornecido. Rapidamente nos apercebemos que o buzzer é apenas capaz de ler e reproduzir frequências e as suas respetivas durações. Ao pesquisar sobre formas de reproduzir sons através de um buzzer, deparámo-nos com um código que dividia a *Empirical March* nas suas respetivas notas musicais e duração, através da função *beep (int note, int duration)*. De maneira a reproduzir este processo para todos os efeitos sonoros, transformámos todos os ficheiros .wav para ficheiros tipo midi para posteriormente os convertermos numa pauta musical a ser lida. Contudo, e devido à sobreposição de sons, as pautas eram demasiadas complexas para leitura. De seguida segue o exemplo da pauta musical para o *off sound*:



Figura 1 – Pauta musical correspondente ao *off sound*

De forma a contornar este problema, gerámos, a partir do *software MATLAB*, o valor da frequência ao longo do tempo para cada um dos sons. No entanto, nenhum dos efeitos sonoros tinha valores constantes de frequência ao longo de intervalos suficientemente grandes para ser perceptível a passagem para notas posteriores de maior/menor frequência. Tentámos ainda desenvolver uma função que permitisse a tradução das retas com inclinação (Fig. 2) de forma a obtermos uma transição mais suave entre frequências.

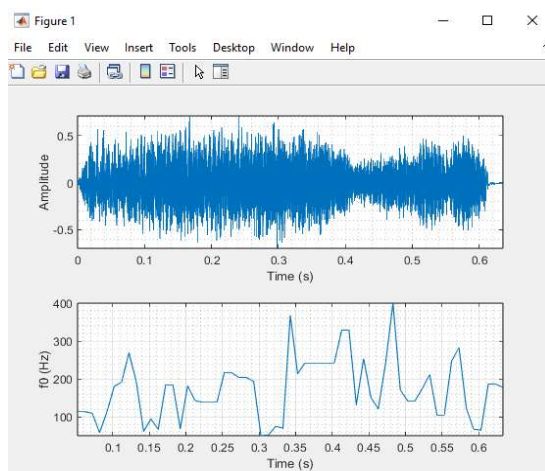


Figura 2 – Gráficos de amplitude e frequência em função do tempo para o som *off sound*

```
void reta(int freq_i, int freq_f, int duration) {
    int freq = freq_i;
    int declive = (freq_f - freq_i)/duration;
    for (int tempo = 0; tempo < duration;
        tempo += duration/10) {
        beep(freq, duration/10);
        freq = freq + declive*tempo;
    }
}
```

Figura 3 – Função *reta* para fazer o fade das frequências

Não satisfeitos com esta execução e muito menos com o som metálico do buzzer, decidimos arranjar uma coluna com  $8\Omega$  e  $1W$ . Tendo-a juntado ao circuito, reparámos que o áudio estava muito baixo e decidimos procurar formas de amplificar o áudio. Depois de 3 tentativas falhadas de montar circuitos com resistências, condensadores e transístores para tentar amplificar o sinal de entrada, resolvemos deixar o áudio padrão.

Contudo, por falta de memória Flash na placa do Arduino, tivemos dificuldades em guardar os sons no programa em si, pelo que tivemos de comprar um módulo para cartões Micro SD, de modo a guardarmos os áudios nele.

Depois de tudo configurado, reparámos que o áudio ficou ligeiramente mais alto, mas ainda abaixo das expectativas. Finalizámos esta parte dessa forma, mas se tivéssemos mais tempo, teríamos arranjado uma forma de amplificar o áudio.

### Circuito Elétrico

Para a montagem do circuito utilizámos uma *breadboard* pequena para condensar o uso de cabos. Nela, existem 2 circuitos: uma resistência de  $1k\Omega$  para cada um dos botões (para resolver o problema da meta-estabilidade do sinal do botão); e uma resistência de  $10\Omega$  e um condensador em série para manter a média das tensões a  $0V$  na coluna.

De resto, é possível observar os componentes todos ligados às suas respectivas portas configuradas no código.

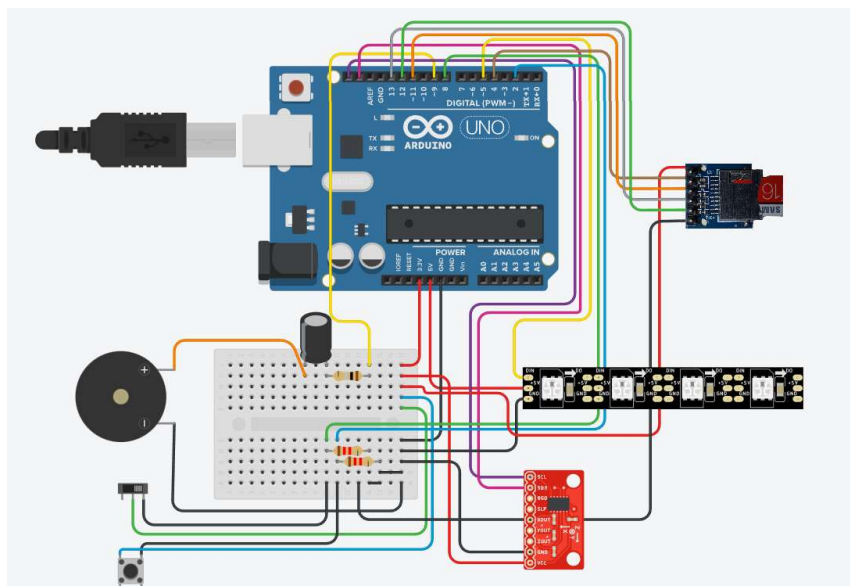


Figura 4 – Ilustração do circuito elétrico através do Tinkercad



## Algoritmo

O algoritmo do código dos componentes funciona simultaneamente e em harmonia no dispositivo. A existência de um botão ON/OFF permite ligar e desligar o *lightsaber* de forma a que todo o código só seja executado quando o botão está no estado ON. Além desse botão, ainda foi colocado um segundo botão que funcionou para alternar entre as várias cores para os LEDs.

A função *setup()* serviu-nos para inicializar os componentes e definir os pinos de leitura e escrita do Arduino para a comunicação.

Já a função *loop()* é a função mais importante do código, pois permite ler os dados do acelerómetro, definir a cor dos LEDs e reproduzir os sons da coluna tudo em paralelo e simultaneamente. Há várias funções que permitem definir qual a cor dos LEDs vai ser emitida, a animação de ligar, a animação de desligar, a decisão dada pelo valor obtido pelo acelerómetro.

Como modo adicional, adicionámos ao *lightsaber* a música característica e icónica da saga do *Star Wars*, *Empirical March*, cuja cor dos LEDs mudava sincronicamente com ritmo da música.

## Protótipo Final



Figura 5– Lightsaber: Protótipo final

## Conclusão

Com a realização deste trabalho pudemos enfrentar um desafio novo que nos forçou a desenvolver novas capacidades e aprender novos conhecimentos em relação a eletrónica, mais especificamente em relação a microcontroladores, sensores e atuadores. Tudo isto nos permitiu chegar a um protótipo que alcançou os parâmetros que idealizámos desde o início. Para além das competências em microcontroladores, desenvolvemos também competências sociais, bem como versatilidade e trabalho em equipa. Mencionamos ainda, a enorme satisfação de nos ter sido permitido a construção de uma réplica de uma saga tão especial a nós!

*May the force be with you.*

## Contribuição

Todos os elementos do grupo trabalharam de igual forma, com empenho e dedicação, pelo que a distribuição da nota no grupo será a seguinte:

- Afonso Lopes (107697) – 25%
- Inês Monteiro (107540) – 25%
- Afonso Vaz (107838) – 25%
- Marta Moniz (107209) – 25%

## Referências

- <https://www.artekit.eu/diy-lightsaber-audio-board/>
- <https://github.com/AlexGyver/EnglishProjects/tree/master/GyverSaber>
- <https://github.com/sidlebj/ECE387Final/wiki>
- <https://randomnerdtutorials.com/guide-for-ws2812b-addressable-rgb-led-strip-with-arduino/>
- [https://github.com/NassimBouyacoub/Star-Wars-theme-Song-Arduino-/blob/master/arduino\\_star\\_wars.ino](https://github.com/NassimBouyacoub/Star-Wars-theme-Song-Arduino-/blob/master/arduino_star_wars.ino)

Obs: Todas os links foram acedidos até à data 20/03/2023. Após esta data, não se garante a disponibilidade de acesso dos mesmos.