Politécnico do Porto

Escola Superior de Media Artes e Design

Carlos Pinto Guedes & José Cunha

Buzzer com controlador

Licenciatura em Tecnologias e Sistemas de Informação Web

Sistemas Computacionais

Orientação: Prof.(ª) Doutor(a) André Baltasar

Vila do Conde, Janeiro de 2018

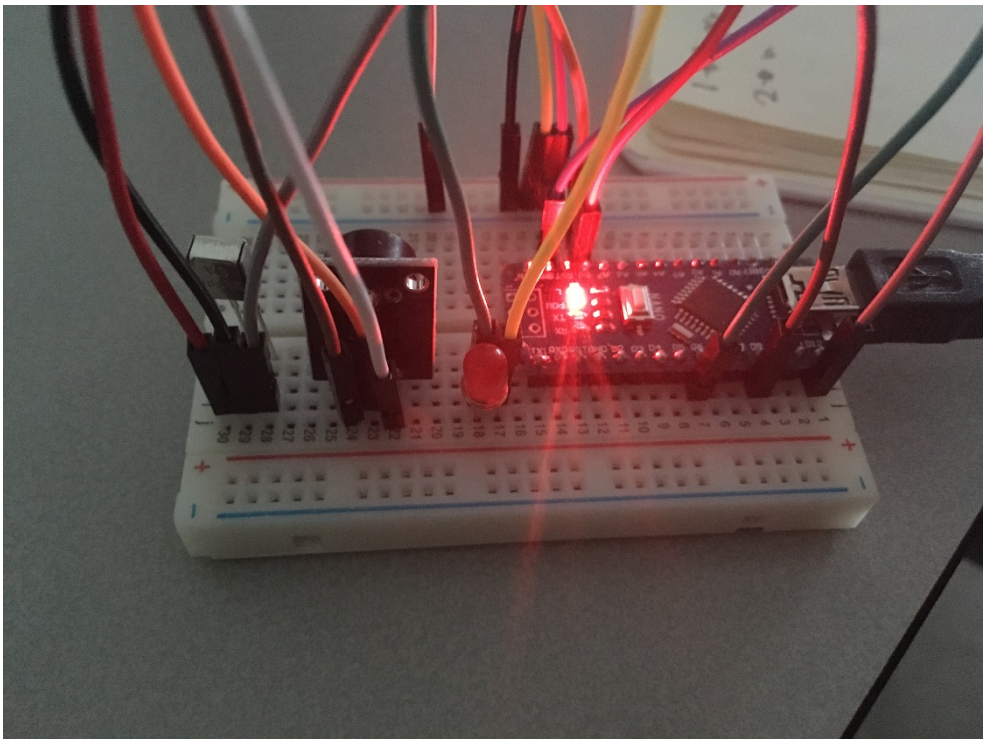
INTRODUÇÃO

Este projeto é diferente da proposta inicial, pois verificamos que não possuíamos os componentes necessários à realização do mesmo.

Sendo assim, mudamos o projeto para “Buzzer com controlador”, onde controlamos um buzzer com o comando para “tocar” músicas (emitir diversos sons que formam uma música).

Para este projeto foi necessário um “buzzer”, um controlador e um recetor de infravermelhos, uma led que pisca ao som da música, de um Arduino Nano, de uma breadboard e de jumpers para conectar tudo.

Aqui está o Arduino com todas as conexões:



1Visão Traseira Arduino

Para que funcionasse, tivemos que apontar todos os dados que as teclas do comando transmitiam ao recetor (neste caso também o Arduino).

Assim, foi uma questão de verificar se o que o recetor recebia, era igual aos códigos já apontados das teclas do comando. Desta maneira, conseguimos iniciar as músicas ao clicarmos numa tecla.

Em relação ao "buzzer", tivemos que definir várias notas (notas utilizadas na música), para que o buzzer as identificasse e tocasse a melodia correta (a nomenclatura das notas está no modo universal – onde o A corresponde ao Lá e assim sucessivamente -).

```
int underworld_melody[] = {
  NOTE_C4, NOTE_C5, NOTE_A3, NOTE_A4,
  NOTE_AS3, NOTE_AS4, 0,
  0,
  NOTE_C4, NOTE_C5, NOTE_A3, NOTE_A4,
  NOTE_AS3, NOTE_AS4, 0,
  0,
  NOTE_F3, NOTE_F4, NOTE_D3, NOTE_D4,
  NOTE_DS3, NOTE_DS4, 0,
  0,
  NOTE_F3, NOTE_F4, NOTE_D3, NOTE_D4,
```

E também tivemos de definir o ritmo da música.

```
int underworld_tempo[] = {
  12, 12, 12, 12,
  12, 12, 6,
  3,
  12, 12, 12, 12,
  12, 12, 6,
  3,
  12, 12, 12, 12,
  12, 12, 6,
  3,
  12, 12, 12, 12,
  12, 12, 6,
  6, 18, 18, 18,
  6, 6,
  6, 6,
  6, 6,
  18, 18, 18, 18, 18, 18,
  10, 10, 10,
  10, 10, 10,
  3, 3, 3
};
```

RESULTADOS

Conseguimos, com sucesso, que o projeto funcionasse.

Explicando:

- ➔ Ao pressionar a tecla “1”, O “buzzer” toca a música “UnderWorld” do famoso jogo Super Mario.
- ➔ Ao pressionar a tecla “2”, o “buzzer” toca a música “Tema do Super Mario”.
- ➔ Ao pressionar a tecla “3”, O “buzzer” toca a música “Pirates of The Caribbean”.
- ➔ Também incluímos o botão de Pausa/Play, onde a qualquer altura para a música.

Obs: Não colocamos mais músicas pois queríamos colocar uma da nossa autoria, mas não tínhamos tempo suficiente para criar uma.

```
/*Acabou ve se entendes*/
if (irrecv.decode(&results)) {
  Serial.println(results.value);
  if (results.value == 16761405) {
    thisNote = size;
  }
  if (results.value == 16718055) {
    thisNote = size;
    sing(2);
  }
  if (results.value == 16743045) {
    thisNote = size;
    sing(3);
  }

  irrecv.resume();
}
```

Esta parte do código é a que identifica a receção de dados vindos do comando, e identifica-os para "tocarem" uma certa música (executarem certa função).

CONCLUSÃO

Este projeto fez-nos sentir mais confortáveis, trabalhando com recetor infra-vermelhos e com buzzer.

Concluímos assim, que o projeto funciona perfeitamente nas condições propostas, mas trabalhando com um “buzzer” não podemos ajustar o volume (na perfeição), onde é uma desvantagem.

Ficamos assim com um simulador de “áudio player”, com controlador via infra-vermelhos.