



---

TRABALHO DE SISTEMAS MICROCONTROLADOS

## PROJETO FINAL

### Open Led Race

Enzo Dornelles Italiano<sup>1</sup>

Henrique Souza Marcuzzo<sup>2</sup>

Matheus Henrique Batistela<sup>3</sup>

CAMPO MOURÃO

DEZEMBRO/2019

---

<sup>1</sup> [enzoitaliano@alunos.utfpr.edu.br](mailto:enzoitaliano@alunos.utfpr.edu.br)

<sup>2</sup> [henriquemarcuzzo@alunos.utfpr.edu.br](mailto:henriquemarcuzzo@alunos.utfpr.edu.br)

<sup>3</sup> [matheusbatistela@alunos.utfpr.edu.br](mailto:matheusbatistela@alunos.utfpr.edu.br)

## Sumário

<b>INTRODUÇÃO</b>	<b>3</b>
<b>MATERIAL</b>	<b>3</b>
<b>METODOLOGIA</b>	<b>4</b>
<b>RESULTADOS E DISCUSSÃO</b>	<b>4</b>
4.1. REALIZAÇÃO	4
4.2. IMPLEMENTAÇÃO	4
4.3. CIRCUITO	5
4.4. DIFICULDADES	6
<b>CONSIDERAÇÕES FINAIS</b>	<b>7</b>
<b>REFERÊNCIAS</b>	<b>7</b>

## 1. INTRODUÇÃO

Este relatório tem como objetivo, escrever a documentação do projeto, incluindo códigos e circuitos auxiliares necessários. Além disso será relatado todas as dificuldades encontradas durante o processo de desenvolvimento do projeto.

O projeto tinha o propósito de diminuir a distância entre o aprendizado da teoria e a prática, melhorar o aproveitamento dos alunos e fomentar o trabalho em equipe. Para isso, seria necessário utilizar todos os conhecimentos adquiridos durante as aulas para que fosse possível cumprir todos os requisitos do projeto, sendo eles: utilizar um display, controlar um led, emitir um som com buzzer ou altofalante, ter um botão ou teclado, se comunicar com algum dispositivo, logar alguma variável de maneira permanente e permitir entradas via porta serial.

## 2. MATERIAL

- 1 LCD texto
- 3 Botões
- 1 Potenciômetro
- 1 Arduino Uno R3
- 1 Real Time Clock RTC DS3231
- 1 Buzer
- 1 Protoboard
- 1 Fita led WS2812B de 5m com 60 leds/m
- 32 Jumpers



Figura 1: Circuito montado

### 3. METODOLOGIA

Por meio de pesquisas em busca de algum projeto já existente que usasse arduino como base, foi encontrada a ideia de montar uma corrida utilizando uma fita led endereçável e foi perceptível o potencial de que esse projeto teria para atingir os requisitos propostos.

O projeto base consistia em um jogo simples de corrida entre dois leds, com um semáforo inicial, a cor do vencedor no final e um som de vitória. Para nossa implementação, criamos um registro de recordes, colocamos um limite para o segundo colocado terminar a corrida em vez de acabar quando o primeiro colocado chegasse, foi feito um tempo de corrida para cada jogador e este assim como o número de voltas de cada jogador passou a ser mostrado no LCD.

### 4. RESULTADOS E DISCUSSÃO

#### 4.1. REALIZAÇÃO

As tarefas para realização do projeto foram bem definidas desde o começo pelo relatório de divisão do trabalho, que tornou tudo mais organizado, fazendo com que fosse possível além de tirar dúvidas sobre eventuais problemas, o grupo conseguisse entregar nos prazos previstos, fazendo com que cada um se comprometesse com o que foi escrito no momento da implementação.

#### 4.2. IMPLEMENTAÇÃO

Embora o código que se encontra disponível no GitHub pelo link <https://github.com/MathBatistela/OpenLedRace>, vale reforçar alguns pontos de como alterar a corrida.

As variáveis que controlam o jogo estão todas concentradas no início do código, deixando para a função setup apenas atividades próprias do arduino.

Logo a primeira variável, que é na verdade uma constante, MAXLED, define qual o número total de leds da fita que estarão sendo usados e consequentemente o tamanho da pista da corrida. O valor máximo que pode ser colocado vai depender e quantos leds existem na sua fita, para este projeto o valor máximo era de 300 leds.

Um pouco mais abaixo existem as constantes COLOR1 e COLOR2, que definem a cor de cada jogar pelo sistema RGB, em que o valor máximo de cada parâmetro é 255, sendo (255,255,255) a cor branca.

Também é possível alterar a música vitória pelo vetor win\_music, existem várias músicas já prontas na internet feitas para arduino utilizando o buzzer, só é necessário trocar os valores para as frequências correspondentes.

O número de voltas necessário para a vitória de um dos jogadores se chama loop\_max, e admite apenas valores positivos e maiores do que zero, estando pré-definida com cinco voltas.

As constantes de aceleração conforme o pressionamento do botão e a desaceleração devido ao atrito estão definidas em ACEL e kf respectivamente, sendo as duas do tipo fracionário e pré-definidas com os valores 0.2 e 0.15 também respectivamente.

Depois das variáveis temos a interrupção que faz a contagem do tempo de cada jogador para posteriormente verificar se o tempo é menor que dos recordes.

E logo após, as funções responsáveis por comandar toda a corrida em todos os aspectos.

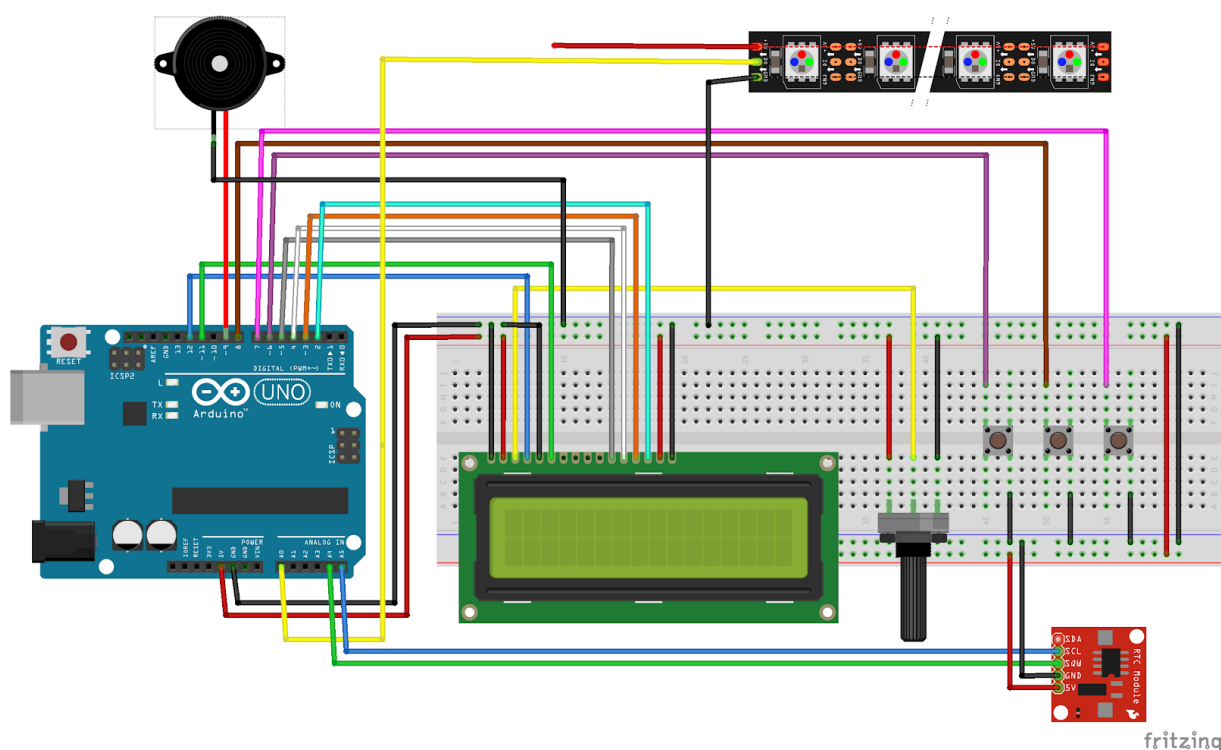
Para começar, como na maior parte dos códigos de arduino existe a função setup, onde são inicializadas as funções do arduino e definições da localização de rampas e loops na pista pelas funções `set_ramp()` e `set_loop()`.

Na função loop, também comum em aplicações de arduino temos uma primeira chamada de inicialização da corrida com a função `start_race()` que é responsável por colorir a pista em certo ponto a fim de formar um semáforo de início de corrida, assim como de atribuir o valor 1 para a flag que identifica se a corrida começou para a outra parte da função loop.

A outra parte do loop que verifica se a corrida começou através da flag, controla tudo na corrida por meio de condicionais, inclusive do desenho dos jogadores na pista. Inicialmente o tempo dos jogadores é atualizado no LCD, e depois é verificado se o jogador 1 ou jogador 2 terminaram a corrida. O número de voltas também é atualizado no LCD pela função `updateTurn()`. A corrida só é finalizada caso os dois jogadores terminem as voltas ou se passarem 10 segundos que o jogador 1 terminou e assim é chamada a função `finish_race()`.

### 4.3. CIRCUITO

A montagem do circuito se deu pela seguinte forma:



### 4.4. DIFICULDADES

O primeiro desafio encontrado foi encontrar o cálculo para tornar o uso da EEPROM automático em relação a captura separada da data e hora retornada pelo RTC.

```

for (i = 0; i < 10; i++){
    Serial.print(i+1); Serial.print(F("\t"));
    int k = 0;
    while (EEPROM.read((50 * i) + k) != '\0'){
        auxChar[0] = EEPROM.read((50 * i) + k);
        Serial.print(auxChar[0]);
        k++;
    }
    Serial.print(F("\t\t\t\t\t"));
    for (j = 0; j < 7; j++){
        auxChar[0] = EEPROM.read(((50 * i) + 33) + j);
        Serial.print(auxChar[0]);
    }
    Serial.print(F("\t\t"));
    for (j = 0; j < 10; j++){
        auxChar[0] = EEPROM.read(((50 * i) + 40) + j);
        Serial.print(auxChar[0]);
    }
    Serial.print(F("\n"));
}

```

O segundo problema encontrado foi por conta do hardware limitado, a falta de memória foi um fator limitante em relação ao uso de variáveis, pois quando o limite era ultrapassado algo não funcionava corretamente, sendo normalmente a fita led. Para resolver foi necessário colocar todas as mensagens para serem armazenadas na memória flash, visto que todas as informações de variáveis são obrigatórias para o funcionamento da corrida.

## 5. CONSIDERAÇÕES FINAIS

Ao final todos puderam observar o cumprimento dos objetivos, visto que eles visavam a integração dos membros e a utilização dos conceitos estudados em sala de aula para melhorar o aproveitamento. Foi possível obter uma fixação melhor do conteúdo e até um aprimoramento do mesmo.

Enfim o objetivo do projeto também foi concluído com a corrida funcionando corretamente, utilizando rampas e loops no meio do circuito, e atendendo aos objetivos propostos pelo regulamento do projeto.

## 6. REFERÊNCIAS

ROSTAN, Gerardo Barbarov. Open LED Race: Minimalist cars race with a strip of smart LEDs and two switches. Hackster.io, mar. 2019. Disponível em: <<https://www.hackster.io/gbarbarov/open-led-race-a0331a#team>>. Acesso em 5 de dez. de 2019.