Robótica Educacional

Módulo 3

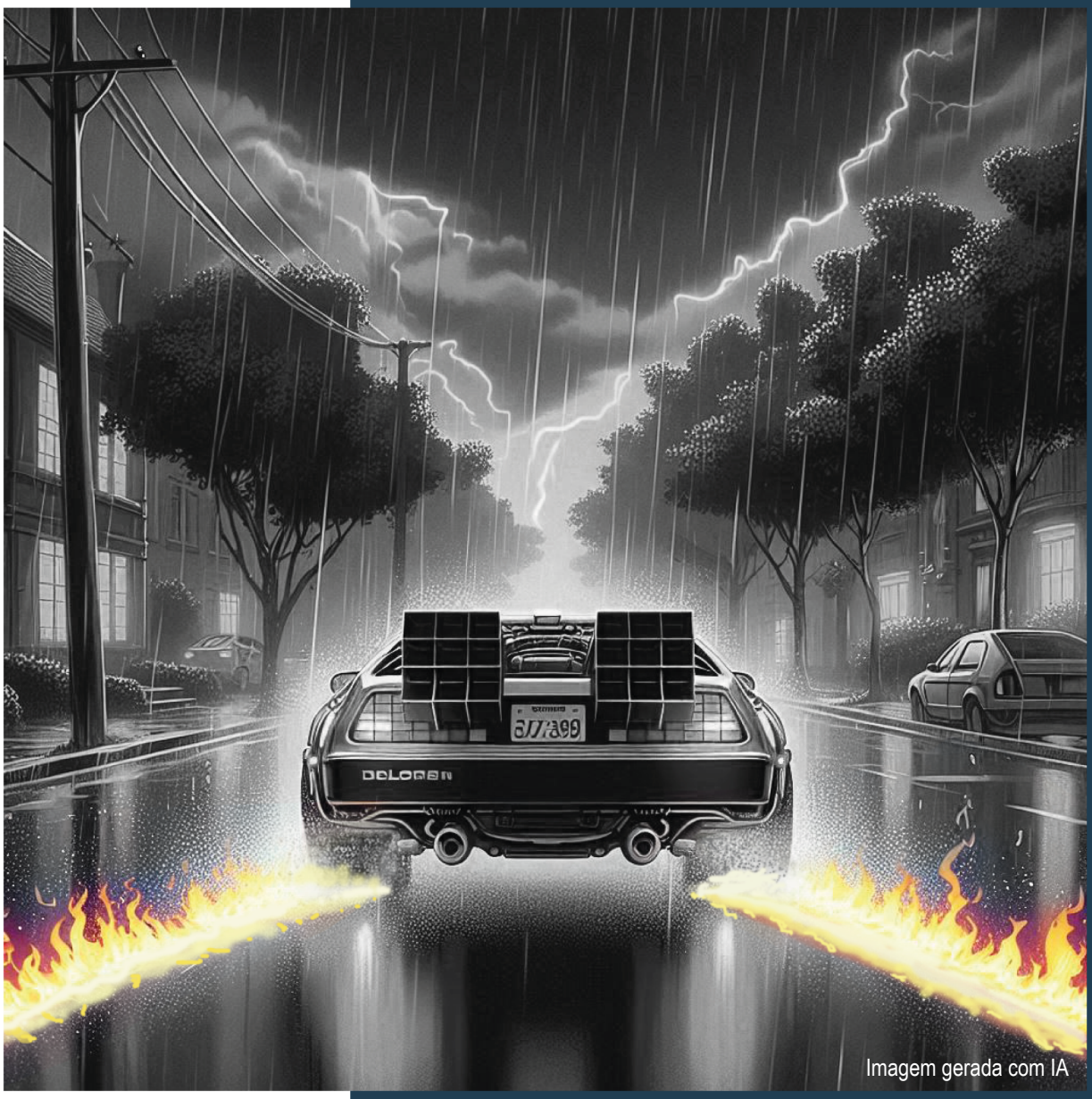


Imagem gerada com IA

Aula 16

De volta para o futuro – II

Diretoria de Tecnologia e Inovação

PARANÁ
GOVERNO DO ESTADO
SECRETARIA DA EDUCAÇÃO

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Ailton Lopes

Darice Alessandra Deckmann Zanardini

Validação de Conteúdo

Cleiton Rosa

Darice Alessandra Deckmann Zanardini

Revisão Textual

Kellen Pricila dos Santos Cochinski

Projeto Gráfico, Diagramação e Geração de Imagem por I.A.

Edna do Rocio Becker

2024

Sumário

Aula 16 De volta para o futuro - II

Introdução

Na última aula, verificamos que o tema "viagem no tempo" já foi bastante abordado na cultura pop em filmes, séries, livros, quadrinhos, etc. A verdade é que a maioria de nós, mais cedo ou mais tarde, gostaríamos de voltar no passado para reviver um momento marcante ou até mesmo avançar para o futuro para saber como serão as coisas. Nesse clima de viagem no tempo, embarcamos na aventura de construir um protótipo do capacitor de fluxo, o dispositivo que permite tal viagem, de acordo com o filme "De Volta para o Futuro". Porém, até agora, fomos capazes de criar 1,21 Gigawatts, verificamos o que acontece de dentro do DeLorean, mas ainda não sabemos o que ocorre lá fora e para onde vamos.

Objetivos desta aula

- Simular o efeito fade out;
- Compreender como funciona a física do rastro de pneus;
- Conhecer um pouco mais da obra "De Volta para o Futuro".

Lista de materiais

- Arduino Uno;
- 2 fitas de LED;
- Protoboard;
- 8 jumpers macho-macho;
- computador ou laptop.

Aula 16 De volta para o futuro - II

Roteiro da aula

1. Contextualização

No filme "De Volta para o Futuro", ao atingir a velocidade de 88mph (aproximadamente 140 km/h), cerca de 1,21 Gigawatts é gerado e o capacitor de fluxo é ativado, o que permite a viagem no tempo.

O Circuito de Tempo é o painel que controla a data para a qual a máquina do tempo vai viajar, mas ele também registra a data atual e a última data em que aconteceu uma viagem no tempo. Por estar configurado de acordo com o sistema americano de registro de horas e datas, o relógio possui dois LEDs que indicam o a.m. (cima) e o p.m. (baixo) marcando antes ou depois do meio-dia. Também é possível perceber que ao mencionar a data, o mês aparece antes por originalmente estar em inglês.

Figura 1 - Réplica do Painel do Circuito de Tempo.

Foto: Wikimedia Commons

Aula 16 De volta para o futuro – II

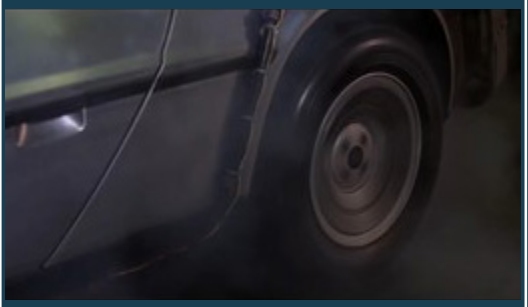
Assista ao trecho onde o Dr. Brown ensina ao Marty McFly como funciona a viagem no tempo:

"De Volta para o Futuro": cachorro Einsten faz uma viagem no tempo | Canal Cena Filmes

Neste mesmo trecho, alguns minutos antes, vamos ver outro detalhe. O Dr. Brown vai mostrar ao Marty McFly como funciona a viagem no tempo, na simulação, o seu cão, Einstein, é o passageiro no veículo DeLorean. Utilizando um controle remoto do tipo que se controla um carrinho, o Dr. consegue manobrar o DeLorean com a distância suficiente para que ele possa atingir a velocidade desejada. O veículo parte em direção dos dois, aumentando a velocidade a cada segundo e alcançando as 88mph e gerando, assim, 1,21 Gigawatts necessários para ativar o capacitor de fluxo e promover a viagem no tempo de acordo com a data estipulada no painel de controle do Circuito de Tempo. Segundos antes de iniciar a viagem, o veículo começa a sofrer algumas descargas elétricas e quando atinge o ápice, desaparece em meio aos dois que estão no estacionamento de um shopping, deixando um rastro de fogo entre as suas pernas. Esse rastro é uma marca indicando que a viagem no tempo aconteceu.

Aula 16 De volta para o futuro - II

Vamos ver quadro a quadro como isso aconteceu:

 DeLorean acelerando no estacionamento	 DeLorean acelerando e pronto para partir	 Momento exato em que atinge os 88mph
 DeLorean quase atingindo os 88mph	 Rastro de fogo entre os dois	 Rastro de fogo deixado pelo DeLorean

Assista o primeiro minuto desse vídeo com a cena:

"De Volta para o Futuro": cena do rastro de fogo deixado pelo DeLorean | Canal Cena Filmes

É justamente esse rastro de fogo que vamos simular! Para isso vamos utilizar a fita de LEDs.

Aula 16 De volta para o futuro – II

Imagem gerada com IA

Por que nos Estados Unidos utilizam Milhas por Hora e não Quilômetros por Hora?

A razão é simples. Os Estados Unidos foram colonizados pela Inglaterra que trouxeram consigo o sistema imperial que incluía a medida da distância em milhas. Com o passar do tempo, essa tradição foi incorporada ao cotidiano americano e mesmo após a independência, continuaram a utilizar tal sistema que até hoje está em vigor. Muitos estrangeiros que decidem viver nos EUA, demoram a se acostumar com o sistema imperial. Entre eles, muitos brasileiros...

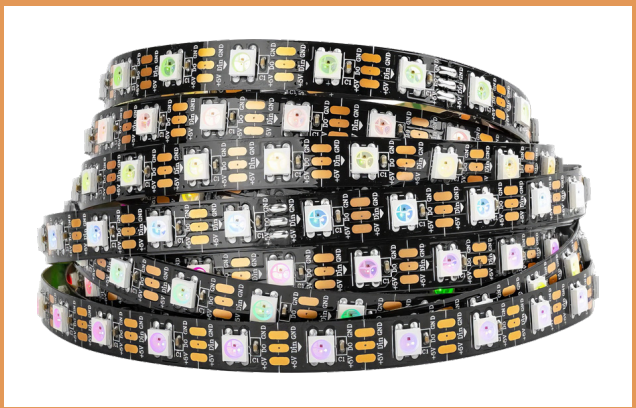
O sistema imperial também tem outras diferenças como as polegadas (comprimento), onças (pesos), galões (volume), Fahrenheit (temperatura), etc.

Aula 16 De volta para o futuro – II

A fita de LEDs é um componente que permite a simulação de diversos efeitos visuais. Disponível em modelos variados quanto à extensão e número de LEDs, as fitas são compostas por um conjunto de LEDs interligados em base flexível, permitindo moldá-las conforme a criatividade e intenção dos projetos.

O modelo presente em nosso kit é a fita **WS2812B** e este conjunto de caracteres possui um significado:

- **W**: refere-se ao fabricante do chip controlador (WorldSemi).
- **S**: indica que é uma fita de LED digital.
- **28**: indica a largura do encapsulamento do chip.
- **12**: representa a quantidade de bits usada para cada cor (vermelho, verde e azul), o que permite uma alta gama de cores e brilho.
- **B**: indica a versão específica do chip, versão atualizada – com melhorias de desempenho e confiabilidade – do modelo original WS2812.



Especificações técnicas

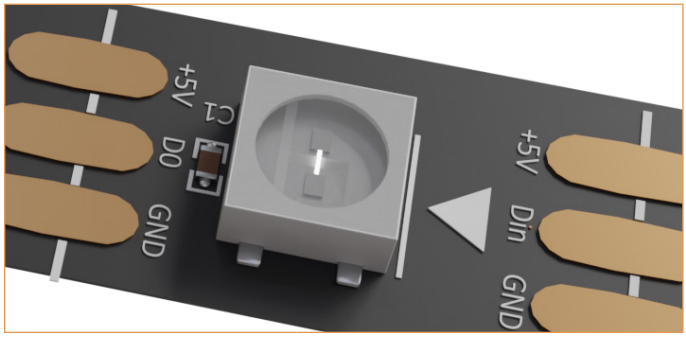
300 LEDs → 5 metros

(60 LEDs / metro)

Tensão de funcionamento: 5 Volts

Potência máxima: 14,4 Watts

Corrente máxima*: 2,88 Amperes



A seta impressa na fita de LEDs mostra a sequência de conexão e acionamento dos LEDs, o que indicamos na programação com o endereço inicial **0**. No caso da nossa fita com 300 LEDs endereçáveis, seu endereço contempla de **0** a **299** e, conforme a quantidade de LEDs em sua fita, você precisará ajustar, na programação, o número total de LEDs (por exemplo, 300) e endereço correspondente (neste caso, 0 a 299).

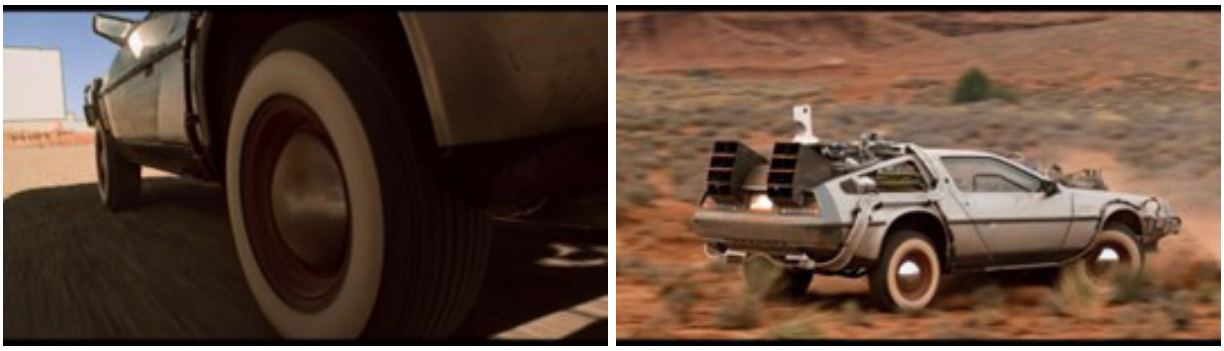
Aula 16 De volta para o futuro - II

2. Montagem e programação

Para simular o efeito do rastro dos pneus do DeLorean, vamos unir forças! Cada kit de robótica contém um rolo de fita LED. Assim, vamos colocar as fitas de LED lado a lado com uma certa distância entre elas. Qual é a distância ideal? Bem, vocês podem decidir entre os grupos.

Você sabia?

Que a cor natural da borracha é branca e não preta como vemos nos pneus dos carros? A borracha fica preta devido à adição de um material chamado negro de fumo que aumenta a durabilidade e resistência ao calor. Nos anos 50 era comum utilizar uma faixa branca nos pneus de carros de luxo, pois eram considerados elegantes. Essa faixa branca era resultado do uso da borracha pura nas laterais. Esse estilo clássico aparece no DeLorean do filme "De Volta para o Futuro III" para representar os veículos da época.



De Volta para o Futuro III (1990)

Caso você possua o protótipo do capacitor de fluxo em mãos, pode utilizá-lo como base para esta montagem. Nossa proposta, considerando o projeto das duas aulas sobre o filme De Volta para o Futuro, é termos um resultado semelhante ao que podemos conferir pelo simulador Wokwi ou pelo nosso modelo 3D.

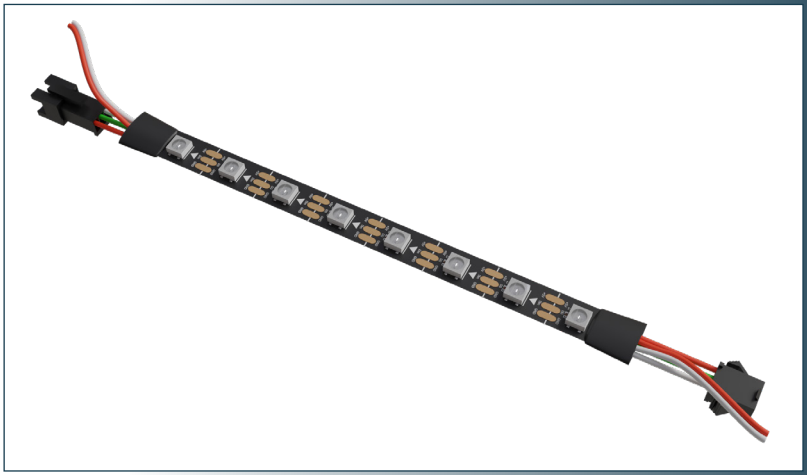
Aula 16 De volta para o futuro – II

Iniciaremos a montagem energizando a Protoboard. Para isso, conecte uma extremidade do jumper macho ao terminal GND da placa Arduino e a outra extremidade na linha negativa (–) livre da Protoboard. Para alimentar a linha positiva (+) livre, utilizaremos a tensão de 5 Volts necessária para fornecer energia para o funcionamento das fitas de LED. Assim, conecte a linha positiva ao terminal 5V da placa Arduino.

Agora vamos alimentar as duas fitas LED na Protoboard. Para isso basta ligar os terminais positivo (vermelho) na linha positiva da placa bem como os terminais negativos (branco) na linha negativa. Se tiver alguma dúvida observe os dizeres 5V (positivo – vermelho) e GND (negativo – branco).

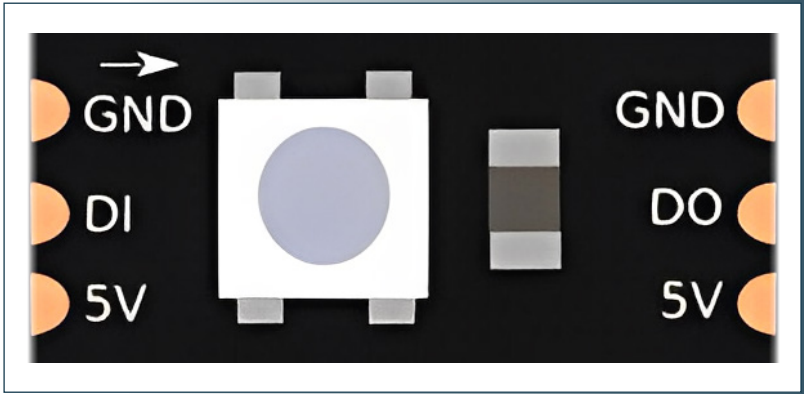
O fio **verde** (DIN – digital input) será ligado nas portas digitais para receber os dados da programação que faremos em breve. No caso do nosso exemplo, utilizaremos as portas **A0** e **A1** – que podem ser indicadas na programação como **14** e **15** – como uma extensão das portas digitais do Arduino, considerando a possibilidade de ampliação do capacitor de fluxo que montamos.

Figura 2 - Fita LED



Fonte: Seed/DTI/CTE.

Figura 3 – Detalhe ilustrativo das conexões da fita LED



Fonte: Seed/DTI/CTE.

Aula 16 De volta para o futuro – II

Agora, vamos programar!

Para a programação do nosso protótipo do rastro, retornaremos tanto à programação da última aula quanto a alguns conteúdos abordados nesse módulo em uma viagem no tempo para ajustes e refinamentos! Assim, mesmo que você não esteja com o capacitor de fluxo montado, a programação no Arduino estará certinha para o projeto completo e quando o relógio da praça soar, todos os efeitos serão ativados!

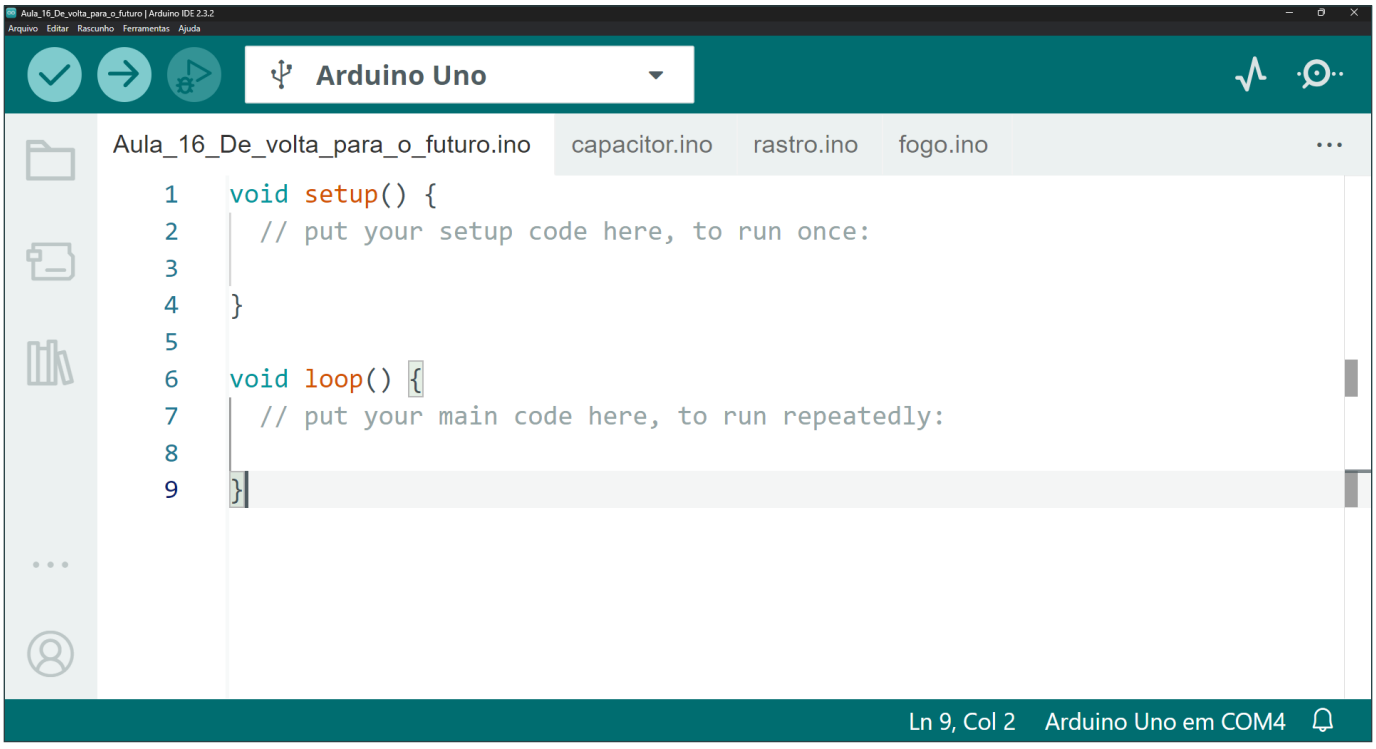
Trabalharemos com o recurso de programação por abas porque para cada efeito criaremos uma função que estará em sua aba específica. Como vimos nas primeiras aulas deste módulo, tanto a criação de funções quanto o recurso de sketch por abas permitem uma melhor organização do código e eventuais ajustes posteriores.

Para facilitar a nossa programação, abra um novo sketch e já crie três abas adicionais:

- **capacitor.ino**
- **rastro.ino**
- **fogo.ino**

Em cada uma dessas abas, criaremos as funções, todas do tipo **void**, referentes ao efeito da viagem no tempo. *Para lembrar como adicionar abas ao sketch de programação e criar funções, consulte a* ***Aula 08 - Técnicas de programação: Sketch com abas***.

Figura 4 - Criação das abas **capacitor.ino**, **rastro.ino** e **fogo.ino**



Fonte: Arduino IDE.

Aula 16 De volta para o futuro - II

Abra a programação da última aula e copie todo o código no sketch principal – ele será o ponto de partida para a nossa programação completa de "De Volta para o Futuro", deixando Dr. eufórico!

Quadro 1 - Programação da Aula 15 - De Volta para o Futuro [Parte I]

```
/* Definição inicial da velocidade do efeito. */

int tempoEfeito = 200;

void setup() {
  /* Define os pinos dos LEDs como saídas. */
  for (int i = 2; i <= 7; i++) {
    pinMode(i, OUTPUT);
  }
}

void loop() {
  /* Liga e desliga os LEDs em sequência. */
  for (int i = 7; i >= 2; i--) {
    digitalWrite(i, HIGH);
    delay(tempoEfeito);
    digitalWrite(i, LOW);
    delay(tempoEfeito);
  }
```

Aula 16 De volta para o futuro – II

```
    /* Reduz o tempo do efeito a cada ciclo. */
    tempoEfeito -= 10;
    if (tempoEfeito == 0) {
      /* Aciona todos os LEDs simultaneamente por 500ms.*/
      for (int i = 7; i >= 2; i--) {
        digitalWrite(i, HIGH);
      }
      delay(500);
      /* Apaga todos os LEDs simultaneamente. */
      for (int i = 7; i >= 2; i--) {
        digitalWrite(i, LOW);
      }
    }
  }
```

Vamos agora ao primeiro ajuste da nossa programação! No sketch principal, manteremos no preâmbulo a criação da variável **tempoEfeito** e, no **void setup()**, a definição dos pinos dos LEDs do capacitor de fluxo.

Todo o conteúdo presente no **void loop()** colaremos na aba **capacitor.ino** como uma nova função do tipo **void**.

Figura 5 - Criação da função **void capacitor()** na aba **capacitor.ino**

Antes de finalizarmos a função **capacitor()**, vamos adicionar uma linha final na programação da aba **capacitor.ino**, após finalizar a condicional de controle **if()** para controle dos LEDs do capacitor, com a nova função **rastro()** que criaremos na próxima aba e será responsável pelo acionamento da fita de LEDs com o efeito inicial do rastro que o DeLorean deixa no asfalto quando realiza sua viagem no tempo.

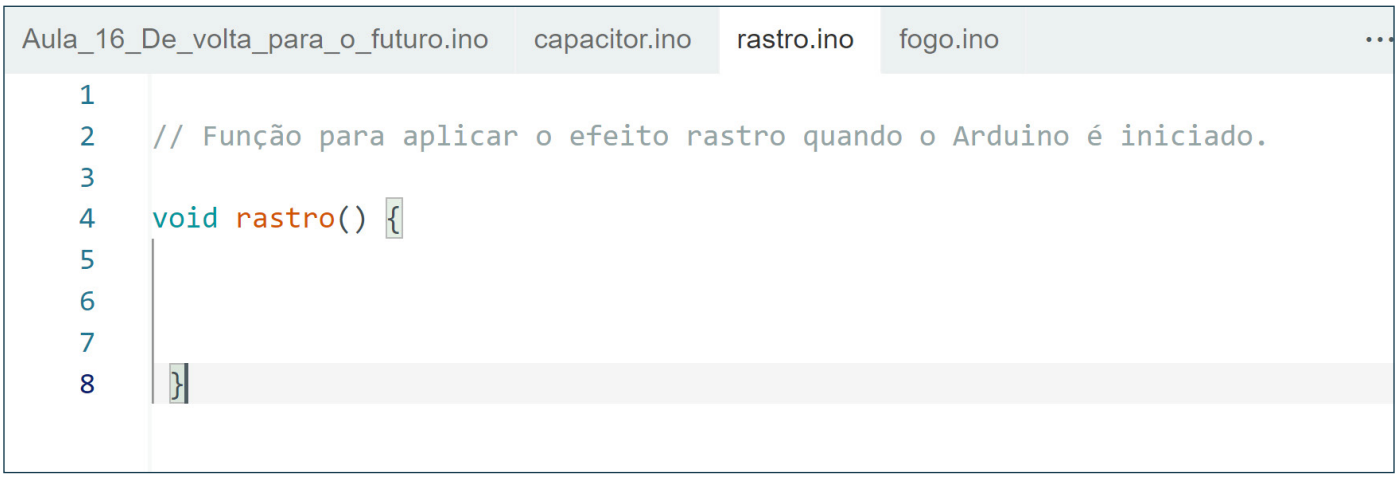
Quadro 2 – Desenvolvimento da função **void capacitor()** na aba **capacitor.ino**

```
/* Função para acionar o capacitor de fluxo. */

void capacitor() {

  /* Liga e desliga os LEDs em sequência. */
  for (int i = 7; i >= 2; i--) {
    digitalWrite(i, HIGH);
    delay(tempoEfeito);
    digitalWrite(i, LOW);
    delay(tempoEfeito);
  }
```

```
  /* Reduz o tempo do efeito a cada ciclo. */
  tempoEfeito -= 10;

  if (tempoEfeito == 0) {
    /* Aciona todos os LEDs simultaneamente por 500ms.*/
    for (int i = 7; i >= 2; i--) {
      digitalWrite(i, HIGH);
    }

    delay(500);
    /* Apaga todos os LEDs simultaneamente. */
    for (int i = 7; i >= 2; i--) {
      digitalWrite(i, LOW);
    }
   }

  /* Chama função para aplicar o efeito rastro quando o Ar-
duino é iniciado. */
  rastro();
}
```

Esta função **rastro()** também é do tipo **void** e vamos criá-la na sua aba específica.

Figura 6 - Criação da função **void rastro()** na aba **rastro.ino**



A partir desse momento, trabalharemos com uma programação específica para o controle das fitas de LED. E para facilitar, utilizaremos a biblioteca **Adafruit NeoPixel**, desenvolvida por Adafruit.

Essa biblioteca possui uma série de funções e comandos que auxiliam na configuração da intensidade dos brilhos dos LEDs da fita e no acionamento de cada um deles, sendo bem versátil nos mais variados projetos.

Dentre os comandos disponíveis na biblioteca, nossa programação completa "De Volta para o Futuro" utilizará os seguintes:

- **.begin()**: inicializa a fita de LEDs.
- **.setBrightness(brilho)**: controla a intensidade dos LEDs da fita.
  - *brilho*: valor entre **0** e **255** para o brilho aplicado aos LEDs da fita.
    - **Atenção:** quando conectada ao Arduino, utilize valores mais baixos de brilho para evitar sobrecargas e danos ao Arduino.
- **.setPixelColor(LED, R, G, B)**: define o acionamento e as cores dos LEDs.
  - *Parâmetros*:
    - **LED**: endereço, iniciando em 0, do LED que se deseja configurar.
    - **R**, **G**, **B**: valores de 0 a 255 para cada uma das cores.
- **.clear()**: desliga todos os LEDs da fita.
- **.show()**: aplica as configurações definidas pelos comandos setPixelColor ou **clear**.

Aula 16 De volta para o futuro – II

Vamos voltar à programação? No preâmbulo da programação principal, adicione a diretiva **#include** para indicar ao Arduino o uso da biblioteca **Adafruit Neopixel**, lembrando de instalá-la caso você utilize a versão software Arduino IDE.

Essa biblioteca exige a criação de **objeto de controle** com dois parâmetros:

- **Adafruit_NeoPixel nome_do_objeto(n_LEDs, porta_fita)**
  - *nome_do_objeto*: atribua um nome para cada fita a ser controlada.
  - *n_LEDs*: informe o número de LEDs presente na fita.
  - *porta_fita*: indique a porta utilizada para conexão de cada fita.

Como nossa programação controlará duas fitas com 300 LEDs em cada e conectadas às portas **A0** (**14**) e **A1** (**15**), criaremos os objetos de controle **fitaLED1** e **fitaLED2** e a variável **int numeroLEDs** = **300** para indicar o número total de LEDs presente em cada fita (caso suas fitas possuam outras quantidades de LEDs, altere esse valor).

Como a biblioteca permite também o controle da intensidade dos LEDs, para não sobrecarregar o Arduino devido à alimentação direta da fita na placa, criaremos a variável **int brilho = 15** para definir o brilho amenizado dos LEDs de cada fita.

Quadro 3 - **Preâmbulo** da programação da Aula 16 – De Volta para o Futuro [Parte II]

```
#include <Adafruit_NeoPixel.h> /* Inclui a biblioteca Adafruit NeoPixel. */

Adafruit_NeoPixel fitaLED1(300, 14); /* Define o número de LEDs na fita e a porta
digital conectada. */

Adafruit_NeoPixel fitaLED2(300, 15); /* Define o número de LEDs na fita e a porta
digital conectada. */

int numeroLEDs = 300; /* Indica o número total de LEDs de cada fita. */

int brilho = 15; /* Define o brilho dos LEDs de 0 a 255. Utilize valores baixos na
fita física conforme alimentação. */

int tempoEfeito = 200; /* Definição inicial da velocidade do efeito aplicado ao ca-
pacitor. */
```

Aula 16 De volta para o futuro – II

Completaremos o **void setup()** da programação desta aula adicionando, para cada objeto de controle criado, os comandos **begin()** e **setBrightness(brilho)** da biblioteca **Adafruit NeoPixe**l, referentes à inicialização desses objetos de controle e à definição da intensidade dos LEDs de cada fita conforme definimos no preâmbulo pela variável **brilho**.

Quadro 4 – **Setup** completo da programação Aula 16 – De Volta para o Futuro [Parte II]

```
void setup() {


  for (int i = 2; i <= 7; i++) {
    pinMode(i, OUTPUT); /* Define os pinos dos LEDs do capacitor de fluxo como saídas. */
  }


  fitaLED1.begin(); /* Inicializa a fita 1. */
  fitaLED2.begin(); /* Inicializa a fita 2. */
  fitaLED1.setBrightness(brilho); /* Define o brilho dos LEDs entre 0 a 255. *Diminuir na fita física. */
  fitaLED2.setBrightness(brilho); /* Define o brilho dos LEDs entre 0 a 255. *Diminuir na fita física. */


}
```

Voltando à aba **rastro.ino**, vamos criar nela a função **void rastro()**?

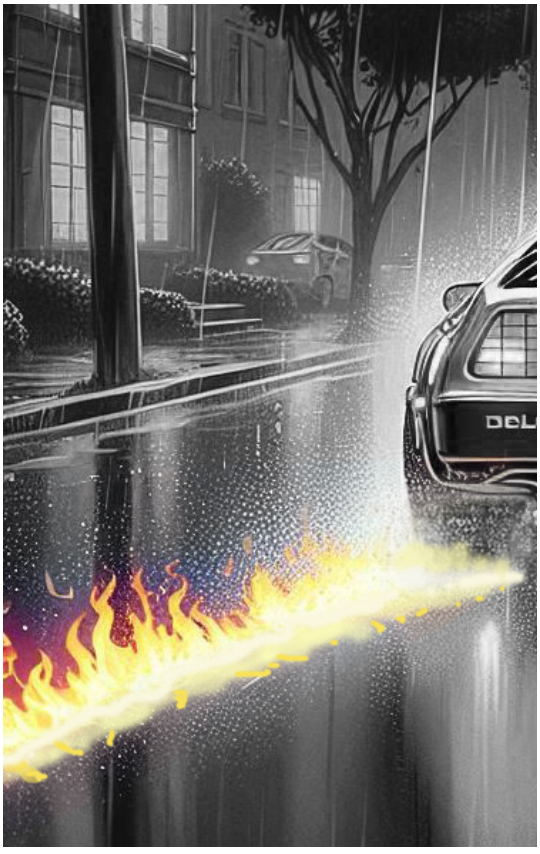
A lógica aplicada nesta função considera o valor armazenado na variável **tempoEfeito** a cada ciclo de acionamento dos LEDs do capacitor de fluxo. Na programação da função **void capacitor()**, cada ciclo de acionamento dos LEDs reduz o valor inicial da variável em 10 milissegundos na linha **tempoEfeito -= 10**.

Aula 16 De volta para o futuro – II

Portanto, definimos na programação da função **rastro()**, chamada durante a execução da função **capacitor()**, que se o valor **tempoEfeito** chegar a **0**, pela condicional **if** aplicaremos um loop controlado para promover o acionamento das fitas de LEDs de modo sequencial.

Quadro 5 - Definição da condição para o rastro

```
/* Função para aplicar o efeito rastro quando o Arduino é iniciado. */
void rastro() {
 if (tempoEfeito == 0) {



 }
}
```



Na sequência, adicionamos o **for()** para percorrer toda a extensão das fitas – por isso, importante definir o número exato de LEDs de cada fita no preâmbulo da programação pela variável **numeroLEDs** para que a repetição percorra todos os LEDs, com a variável **i** iniciando em **0** (primeiro LED de cada fita) e, a cada iteração, seguindo de um em um até completar a totalidade da fita, quando **i** corresponderá ao valor **300**. Então, aplicamos os comandos **setPixelColor()** e **show()** para acionamento e exibição dos LEDs de modo sequencial – considerando os dois objetos de controle **fitaLED1** e **fitaLED2** – e atribuímos a cor laranja a cada LED representado pela variável **i** com os valores RGB **255, 170, 0**. Finalizamos o **for()** com a pausa de **20 milissegundos** entre cada acionamento, o que cria um efeito gradual de exibição do laranja ao longo da fita.

Quadro 6 - Atribuição do **for()** ao **if()** da função rastro

```
 for (int i = 0; i < numeroLEDs; i++) {  /* Loop para percorrer todos os LEDs das fi-
tas. */
   fitaLED1.setPixelColor(i, 255, 170, 0);  /* Acionamento dos LEDs da fita 1 na cor
laranja. */
   fitaLED2.setPixelColor(i, 255, 170, 0); /* Acionamento dos LEDs da fita 2 na cor
laranja. */
   fitaLED1.show();  /* Aplica a modificação na fita 1. */
   fitaLED2.show();  /* Aplica a modificação na fita 2. */
   delay(20);  /* Pausa entre os acionamentos.*/
}
```

Voltando à cena do filme, o que ocorre quando o DeLorean parte? Além do rastro surgir, ele fica por um tempo no asfalto com efeito de fogo. Então fecharemos a programação da função **void rastro()** chamando a nova função **fogo()**, a qual criaremos na sequência.

Quadro 7 - Função **void rastro()** completa

```
/* Função para aplicar o efeito rastro quando o Arduino é iniciado. */
void rastro() {
 if (tempoEfeito == 0) {


 for (int i = 0; i < numeroLEDs; i++) {  /* Loop para percorrer os LEDs.*/
   fitaLED1.setPixelColor(i, 255, 170, 0);  /* Acionamento dos LEDs da fita 1 na cor
laranja.*/
   fitaLED2.setPixelColor(i, 255, 170, 0); /* Acionamento dos LEDs da fita 2 na cor
laranja.*/
   fitaLED1.show();  /* Aplica a modificação na fita 1.*/
```

```
  fitaLED2.show();  /* Aplica a modificação na fita 2.*/
  delay(20);  /* Pausa entre os acionamentos.*/
 }

/* Chama função para aplicar o efeito fogo após inicialização com o rastro.*/
 fogo();
 }
}
```

Visando a organização do nosso código, também criaremos a função **void fogo()** na sua aba específica, **fogo.ino**.

Figura 7 - Criação da função **void fogo()** na aba **fogo.ino**

Aula_16_De_volta_para_o_futuro.ino | capacitor.ino | rastro.ino | fogo.ino

```
1  // Função para aplicar o efeito fogo após o rastro.
2
3  void fogo() {
4
5
6
7
8  }
```

Para essa função, nosso objetivo é atribuir às fitas de LEDs um efeito de fogo por **5 segundos**, que desaparecerá gradualmente. Para isso, utilizaremos a função **millis()**. *Para saber mais, acesse a **Aula 09 - Função millis( )**.*

Aula 16 De volta para o futuro - II

Na aba **fogo.ino**, iniciamos a função **void fogo()** criando uma variável do tipo **unsigned long** com a função **millis()**, o que nos permite saber o tempo decorrente desde o início do efeito que será executado por **5 segundos**.

Quadro 8 - Criação da variável para armazenar os dados de leitura da **função millis()**

```
/* Função para aplicar o efeito fogo após o rastro. */


void fogo() {
 /* Armazena o tempo inicial. */
  unsigned long tempoInicial = millis();


}
```

Como a variável **tempoEfeito** guardará o tempo decorrido do Arduino, nossa proposta é que enquanto ela possuir um valor igual ou menor a **5000 milissegundos**, o efeito fogo seja aplicado às fitas de LEDs. Portanto, adicionaremos à nossa função um **while()** para termos a alternância das cores azul e laranja durante este período após a função **fogo()** ser chamada quando o rastro de acionamento dos LEDs for concluído.

Para a alternância entre as cores, aplicaremos também os comandos **setPixelColor()** e **show()** a cada objeto de controle, porém para que o acionamento dos LEDs seja simultâneo e não sequencial como na função **rastro()**, tanto o comando **show()** quanto a função **delay()** entre cada acionamento estarão fora do loop de repetição **for()**. Deste modo, o **for()** primeiro percorrerá todos os LEDs e depois atribuirá as cores azul e laranja com um intervalo **50 milissegundos** entre cada alteração de cor, definidas pelos valores **RGB 0, 0, 200** e **255, 170, 0**.

Quadro 9 - Atribuição e transição entre as cores azul e laranja na fita de LEDs

```
while (millis() - tempoInicial < 5000) {  /* Executa o efeito fogo alternando as co-
res por 5 segundos.*/

   for (int i = 0; i < numeroLEDs; i++) {  /* Loop para percorrer os LEDs.*/
    fitaLED1.setPixelColor(i, 0, 0, 200);  /* Acionamento dos LEDs da fita 1 na cor
azul.*/
    fitaLED2.setPixelColor(i, 0, 0, 200);  /* Acionamento dos LEDs da fita 2 na cor
azul.*/
   }

   fitaLED1.show();  /* Aplica a modificação na fita 1.*/
   fitaLED2.show();  /* Aplica a modificação na fita 2.*/
   delay(50);  /* Pausa entre os acionamentos.*/

   for (int i = 0; i < numeroLEDs; i++) {  /* Loop para percorrer os LEDs.*/
    fitaLED1.setPixelColor(i, 255, 170, 0); /* Acionamento dos LEDs da fita 1 na cor
laranja.*/
    fitaLED2.setPixelColor(i, 255, 170, 0);  /* Acionamento dos LEDs da fita 2 na
cor laranja.*/
   }

   fitaLED1.show();  /* Aplica a modificação na fita 1.*/
   fitaLED2.show();  /* Aplica a modificação na fita 2.*/
   delay(50);  /* Pausa entre os acionamentos.*/

 }
```

Aula 16 De volta para o futuro – II

Na sequência da transição das fitas entre azul e laranja por 5 segundos, como definimos pelo **while()** que considera o valor armazenado na variável **tempoEfeito**, teremos um novo loop controlado para um efeito **fade-out**.

Esse novo loop decrementa gradualmente os valores da cor atribuída a cada fita até que todos os valores de **RGB** estejam em **0** e nos tragam o efeito de que o rastro do DeLorean sumiu. Portanto, teremos para o comando **setPixelColor** a utilização de duas variáveis de controle com **for()**:

- **cor**, para **decrementar** os valores **R** e **G** que definem o amarelo inicial.

```
for (int cor = 255; cor >= 0; cor--)
```

- **i**, para percorrer todos os LEDs das fitas antes de aplicar os efeitos e atribuir um delay de 10 milissegundos para o efeito gradual de desaparecimento do rastro.

```
for (int i = 0; i < numeroLEDs; i++)
```

Quadro 10 - Aplicação do efeito fade-out no rastro de fogo

```
/* Aplica efeito fade-out amarelo para apagar o rastro. */
for (int cor = 255; cor >= 0; cor--) { /* Loop para decrementar os valores de cor RGB.*/
  for (int i = 0; i < numeroLEDs; i++) { /* Loop para percorrer os LEDs.*/
    fitaLED1.setPixelColor(i, cor, cor, 0); /* Acionamento dos LEDs da fita 1 com decremento da cor.*/
    fitaLED2.setPixelColor(i, cor, cor, 0); /* Acionamento dos LEDs da fita 2 com decremento da cor.*/
  }
  fitaLED1.show(); /* Aplica a modificação na fita 1.*/
  fitaLED2.show(); /* Aplica a modificação na fita 2.*/
  delay(10); /* Ajuste o tempo do fade.*/
}
```

Aula 16 De volta para o futuro – II

Com esses recursos de controle, finalizamos também a programação da nossa função **void fogo()**, completando todos os efeitos relacionados à viagem do DeLorean.

Quadro 11 - Função **void fogo()** completa

```
/* Função para aplicar o efeito fogo após o rastro.*/

void fogo() {
  /* Armazena o tempo inicial.*/
  unsigned long tempoInicial = millis();

  while (millis() - tempoInicial < 5000) {  /* Executa o
efeito fogo alternando as cores por 5 segundos.*/

    for (int i = 0; i < numeroLEDs; i++) {  /* Loop para per-
correr os LEDs.*/
      fitaLED1.setPixelColor(i, 0, 0, 200);  /* Acionamento
dos LEDs da fita 1 na cor azul.*/
      fitaLED2.setPixelColor(i, 0, 0, 200);  /* Acionamento
dos LEDs da fita 2 na cor azul.*/
    }

    fitaLED1.show();  /* Aplica a modificação na fita 1.*/
    fitaLED2.show();  /* Aplica a modificação na fita 2.*/
    delay(50);  /* Pausa entre os acionamentos.*/
```

```
    for (int i = 0; i < numeroLEDs; i++) {  /* Loop para per-
correr os LEDs.*/
      fitaLED1.setPixelColor(i, 255, 170, 0);  /* Acionamento
dos LEDs da fita 1 na cor laranja.*/
      fitaLED2.setPixelColor(i, 255, 170, 0);  /* Acionamento
dos LEDs da fita 2 na cor laranja.*/
    }

    fitaLED1.show();  /* Aplica a modificação na fita 1.*/
    fitaLED2.show();  /* Aplica a modificação na fita 2.*/
    delay(50);  /* Pausa entre os acionamentos.*/

  }

  /* Aplica efeito fade-out amarelo para apagar o rastro.
  for (int cor = 255; cor >= 0; cor--) {  /* Loop para decre-
mentar os valores de cor RGB.*/
    for (int i = 0; i < numeroLEDs; i++) { /* Loop para per-
correr os LEDs.*/
      fitaLED1.setPixelColor(i, cor, cor, 0); /* Acionamento
dos LEDs da fita 1 com decremento da cor.*/
      fitaLED2.setPixelColor(i, cor, cor, 0); /* Acionamento
dos LEDs da fita 2 com decremento da cor.*/
    }
    fitaLED1.show();  /* Aplica a modificação na fita 1.*/
    fitaLED2.show();  /* Aplica a modificação na fita 2.*/
    delay(10);  /* Ajuste o tempo do fade.*/
  }
}
```

Aula 16 De volta para o futuro – II

Confira a programação completa do nosso protótipo, contemplando o capacitor de fluxo e as fitas de LEDs:

Protótipo com programação completa

Boa viagem!

Imagem gerada com IA

Aula 16 De volta para o futuro - II

Desafios:

Você e seus colegas podem também integrar um display OLED ao protótipo para simular o odômetro com a aceleração do DeLorean quando ele atingir 88mph.

Que tal vocês ampliarem o projeto e criar também o painel do DeLorean? Você e seus colegas podem pensar na inserção de componentes para a definição e registro de datas.

E se...

O projeto não funcionar?

- Verifique as conexões e atribuição das portas utilizadas no protótipo.
- Confira se programação final do seu projeto contempla todas as funções criadas para cada efeito aplicado.
- Verifique a criação de um objeto de controle para cada fita LED conectada, com a conferência das portas utilizadas, e se os comandos da biblioteca Adafruit NeoPixel são aplicados a ambos os objetos de controle.

Dica!

Você pode conectar também as duas fitas de LEDs à mesma porta digital e utilizar apenas um objeto de controle no decorrer da programação para o controle de ambas ligadas à mesma porta.

- Confira a polaridade dos componentes utilizados no projeto.
- Revise a programação quanto à intensidade aplicada a cada fita LED.

3. Feedback e finalização

O tema "viagem no tempo" gera um fascínio e envolve muitas expectativas e projeções. Como foi, para você e seus colegas, reproduzirem uma das cenas mais icônicas do cinema sobre a temática?

Nossa sequência de aulas sobre o filme "De Volta para o Futuro" foi uma verdadeira aventura: recriamos o momento em que o DeLorean acelera até 88mhp e desaparece em um rastro de fogo. Como vocês perceberam, não há limites para a imaginação e, em vez de plutônio e fluxo temporal real, utilizamos LEDs avulsos para o capacitor de fluxos da **Aula 15** e em fitas para o rastro de fogo desta aula, com o controle desses componentes de modo a piscarem em um ritmo definido e alterarem as cores até desaparecem, como se o DeLorean estivesse partindo para outra época!

Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado e lembrem-se que o futuro não está escrito, ele depende de nós!

Bons projetos!! E não se esqueçam de recarregar seus capacitores de fluxo para as próximas aventuras!

REFERÊNCIAS

ARDUINO. **Documentação de Referência da Linguagem Arduino**. Disponível em: https://www.arduino.cc/reference/pt/. Acesso em: 27 mai. 2024.

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)

COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

- Adilson Carlos Batista
- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva

Os materiais, aulas e projetos da "Robótica Paraná", foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica.
Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.

DTI - DIRETORIA DE TECNOLOGIA E INOVAÇÃO

PARANÁ
GOVERNO DO ESTADO
SECRETARIA DA EDUCAÇÃO