

Instructions for Authors of SBC Conferences Papers and Abstracts

**Thaiane da Silva Alves Ribeiro¹, Leonardo Noronha de Andrade², Lucas Escobar³,
Daniela Rocha⁴**

¹Bacharelado Sistemas de Informação – Faculdade Dom Bosco de Porto Alegre
(FDBPOA)
90.520-280 – Porto Alegre – RS – Brazil

{

Abstract. . *This article presents the development and assembly of a led mask, using Arduino and other materials, the group also needed knowledge in electronics and soldering of materials to connect the components and reach the final result.*

Resumo. *Este artigo apresenta o desenvolvimento e a montagem de uma máscara led, utilizando Arduino e outros materiais, o grupo também necessitou de conhecimento em eletrônica e soldagem de materiais para conectar os componentes e chegar ao resultado final.*

1. Introdução

Neste artigo, vamos explicar o processo de criação de uma máscara com LEDs endereçáveis. Essas máscaras são uma forma criativa e personalizada de expressar sua criatividade e adicionar um toque especial aos seus trajes. Com um conjunto de materiais acessíveis e um pouco de conhecimento em eletrônica, você pode criar sua própria máscara luminosa. Vamos aos detalhes.

Materiais necessários:

- 1 x Arduino Uno
- 1 x Fita LED (144LED + IP65)
- 1 x 134N3P (Carregador de bateria USB)
- 1 x Microfone
- 1 x Li-Ion Bateria
- 1 x Conector (Macho – Fêmea)
- 1 x Máscara

Você pode encontrar esses itens em lojas especializadas em eletrônica ou até mesmo em plataformas de venda online, como o Mercado Livre.

2. Passo-a-passo do processo de montagem

1. Fita de LED endereçável: É crucial utilizar uma fita de LED endereçável, pois ela permite o controle individual dos LEDs. Diferentemente das fitas RGB comuns, as fitas endereçáveis possuem um chip controlador em cada LED, permitindo que o Arduino se comunique com cada LED separadamente.
2. Bateria: É possível substituir a bateria por outras soluções, como uma power bank. Caso opte por utilizar bateria, é importante ter um circuito controlador de carga e step-up de tensão para garantir o funcionamento adequado do sistema. Evite conectar a fita de LED diretamente ao Arduino, pois isso pode exceder a corrente suportada pelo Arduino e danificar o controlador de tensão.

Agora vamos ao processo de criação:

1. Transformando a fita em uma matriz de LEDs: Essa é a etapa mais complexa do projeto. Se possível, adquira uma matriz de LEDs pronta para facilitar o processo. Caso contrário, você precisará cortar e soldar a fita de LEDs para formar a matriz desejada. Certifique-se de seguir as indicações de corte presentes na fita e mantenha a direção correta dos LEDs ao realizar as conexões em formato de zigue-zague.
2. Montagem da matriz: Cole a fita de LEDs em uma base de fita isolante, formando um retângulo. Você também pode colá-la em um retângulo de tecido.
3. Conexão dos componentes eletrônicos: Conecte os fios nos pontos de entrada e saída da fita de LEDs, conforme indicado pelo fabricante. Em seguida, conecte o Arduino Nano à fita de LEDs. Utilize fios para fazer as conexões necessárias, garantindo uma boa fixação e evitando curtos-circuitos.
4. Integração do módulo detector de som: Conecte o módulo detector de som (Max 4466) ao Arduino Nano. Utilize fios para conectar a saída de áudio do módulo ao pino de entrada analógica do Arduino. Isso permitirá que a máscara responda a diferentes níveis de som ambiente.
5. Conexão da bateria: Conecte o circuito para controle de carga da bateria e step-up de 5V ao Arduino Nano e à bateria LiPo. Siga as instruções do circuito para fazer as conexões corretamente. Certifique-se de conectar a bateria de forma segura, evitando curtos-circuitos ou danos aos componentes eletrônicos.
6. Fixação da máscara de pano: Fixe a matriz de LEDs na máscara de pano, garantindo que ela fique posicionada adequadamente para iluminar a área desejada. Você pode costurar a matriz na máscara ou utilizar adesivos fortes para fixá-la.

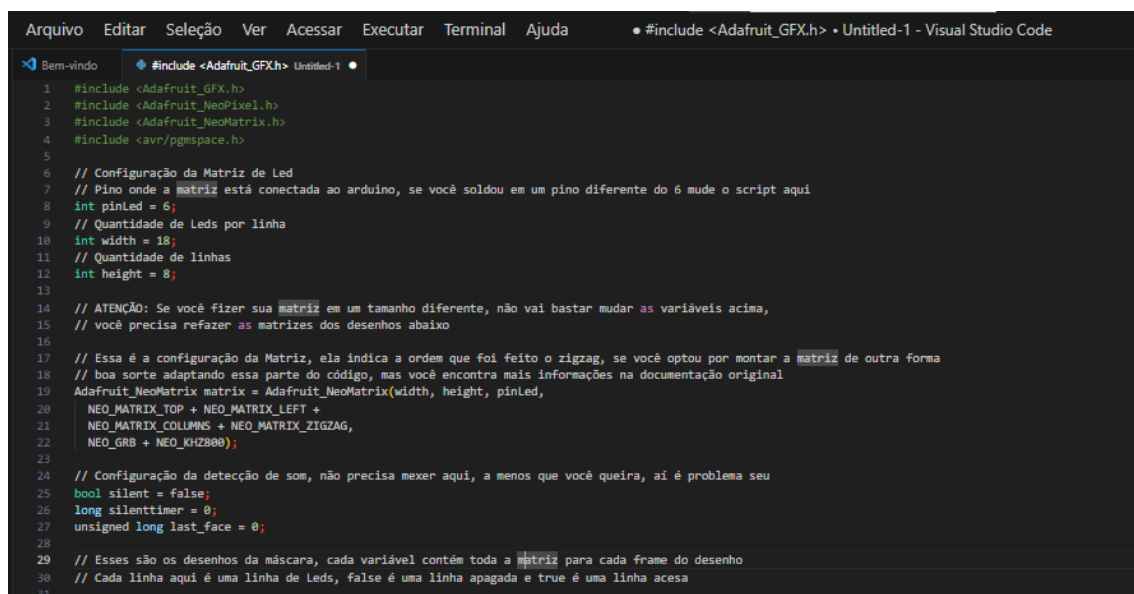
7. Programação do Arduino: Escreva um código Arduino para controlar os LEDs endereçáveis. Existem várias bibliotecas disponíveis que facilitam o controle dos LEDs. Você pode programar diferentes padrões de iluminação, cores e efeitos com base nos dados recebidos do módulo detector de som.
8. Teste e ajustes: Após concluir a montagem e programação, teste a máscara para garantir que os LEDs estão funcionando corretamente. Faça ajustes se necessário, tanto na montagem física quanto no código, para alcançar o resultado desejado.

3. Segurança

Tomamos precauções de segurança ao trabalhar com eletrônicos e baterias. Certifique-se de seguir as especificações dos componentes utilizados e evitar situações que possam causar danos ou riscos.

4. Código completo

Configuração:



```
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda • #include <Adafruit_GFX.h> • Untitled-1 - Visual Studio Code
Bem-vindo • #include <Adafruit_GFX.h> Untitled-1 •
1 #include <Adafruit_GFX.h>
2 #include <Adafruit_NeoPixel.h>
3 #include <Adafruit_NeoMatrix.h>
4 #include <avr/pgmspace.h>
5
6 // Configuração da Matriz de Led
7 // Pino onde a matriz está conectada ao arduino, se você soldou em um pino diferente do 6 mude o script aqui
8 int pinLed = 6;
9 // Quantidade de Leds por linha
10 int width = 18;
11 // Quantidade de linhas
12 int height = 8;
13
14 // ATENÇÃO: Se você fizer sua matriz em um tamanho diferente, não vai bastar mudar as variáveis acima,
15 // você precisa refazer as matrizes dos desenhos abaixo
16
17 // Essa é a configuração da Matriz, ela indica a ordem que foi feito o zigzag, se você optou por montar a matriz de outra forma
18 // boa sorte adaptando essa parte do código, mas você encontra mais informações na documentação original
19 Adafruit_NeoMatrix matrix = Adafruit_NeoMatrix(width, height, pinLed,
20     NEO_MATRIX_TOP + NEO_MATRIX_LEFT +
21     NEO_MATRIX_COLUMNS + NEO_MATRIX_ZIGZAG,
22     NEO_GRB + NEO_KHZ800);
23
24 // Configuração da detecção de som, não precisa mexer aqui, a menos que você queira, aí é problema seu
25 bool silent = false;
26 long silenttimer = 0;
27 unsigned long last_face = 0;
28
29 // Esses são os desenhos da máscara, cada variável contém toda a matriz para cada frame do desenho
30 // Cada linha aqui é uma linha de Leds, false é uma linha apagada e true é uma linha acesa
31
```

Boca fechada e aberta:

Função responsável por desenhar a matriz:

```
};  
// Essa função é responsável por desenhar algum dos desenhos na matriz  
void drawImage(short image_addr){  
  // Limpa a matriz  
  matrix.clear();  
  
  // Loop para ir em cada posição e desenhar pixel a pixel  
  for(int x = 0; x<width; x++){  
    for(int y = 0; y<height; y++){  
      bool light = pgm_read_byte(image_addr+x*y*width);  
  
      // Acende se for pra acender  
      if (light) {  
        // em matrix.Color você pode mudar a cor do led cada parâmetro é uma cor de RGB, os valores vão de 0 a 255, nesse exemplo ele acende em branco  
        matrix.drawPixel(x, y, matrix.Color(000,0,200));  
      }  
    }  
  }  
  
  // Recarrega a matrix com o novo desenho  
  matrix.show();  
}
```

Setup para executar o código:

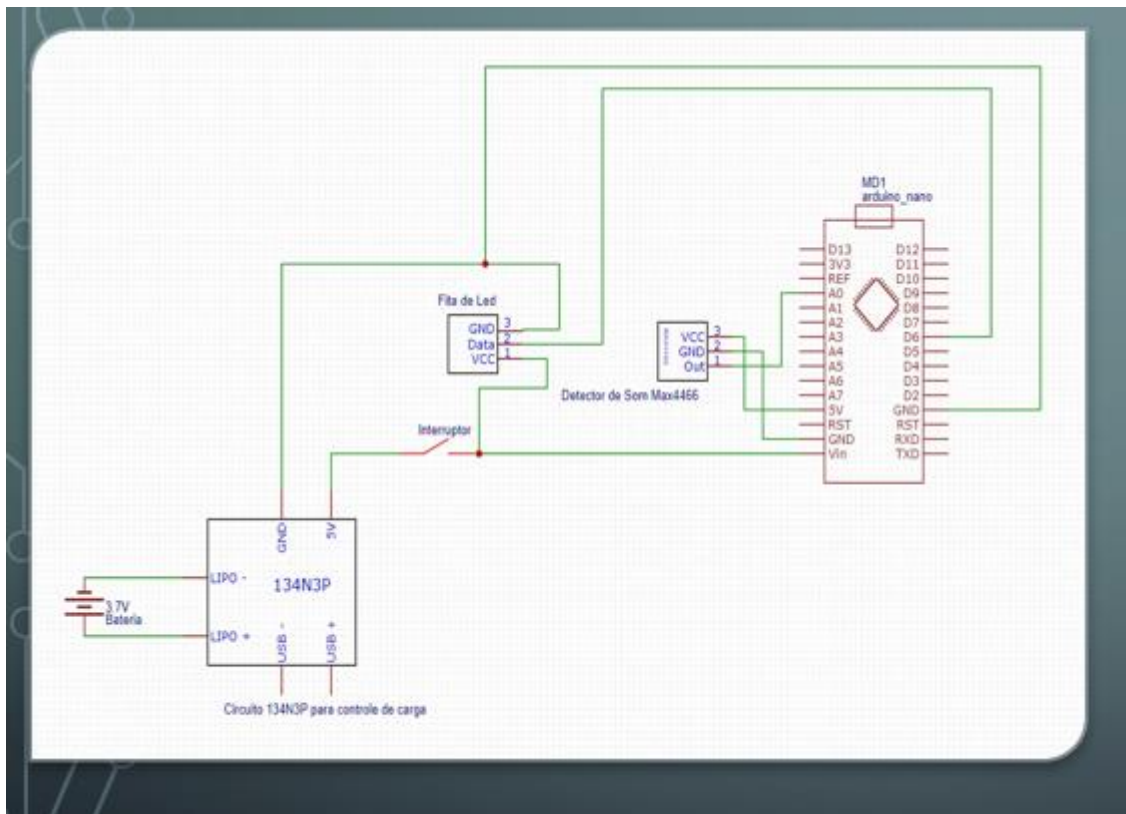
```
24  
25 void setup() {  
26   // O setup prepara tudo para rodar  
27  
28   // Inicializa a matriz NeoPixel  
29   matrix.begin();  
30  
31   // Configura o brilho, uma atenção o brilho vai de 0 a 255, eu achei 50 mais que o suficiente para aparecer bem,  
32   // mas se o tecido da sua máscara for muito grosso pode ser que precise aumentar, mas uma dica, não aumente muito  
33   // se a luz for muito forte o circuito pode não dar conta e ele desligar, além de que com um brilho maior os leds tendem  
34   // a esquentar demais, eu achei 50 confortável  
35   matrix.setBrightness(60);  
36  
37   // Configuro o serial, assim da pra ver a medida de volume do microfone via terminal e ajusta-lo  
38   Serial.begin(9600);  
39 }  
40  
41 float vol = 0;  
42 const uint16_t samples = 128;  
43  
44 void loop() {  
45   // Toda essa parte abaixo é responsável por detectar variação de volume e guardar o valor em uma variável  
46   float nvol = 0;  
47   int previous_peak = -1;  
48  
49   for (int i = 0; i<samples; i++){  
50     auto analog = analogRead(A0);  
51     auto micline = abs(analog - 512);  
52  
53     nvol = max(micline, nvol);  
54   }  
55  
56   vol = (nvol + 1.0*vol)/2.0;
```

Tivemos de instalar a biblioteca "Adafruit NeoPixel" na Arduino IDE antes de carregar o código. Para fazer isso, seguimos o seguinte processo em "Sketch" -> "Incluir Biblioteca" -> "Gerenciar Bibliotecas", procure por "Adafruit NeoPixel" e clique em "Instalar".

Após carregar o código no Arduino, selecionamos a placa e porta corretas antes de fazer o upload.

Após desconectar a USB do Arduino e ligar a máscara, ela começará a exibir o efeito de iluminar todos os LEDs em azul repetidamente. Podemos modificar o código e experimentar diferentes efeitos e cores para personalizar a máscara de acordo com suas preferências.

4.1. Circuito do projeto



4.2. Demonstração funcionamento

Segue vídeo de demonstração da máscara em funcionamento: [MASCARA LED](#)

5. Imagens dos materiais



Referências

<https://medium.com/@adrianomourap/fazendo-uma-m%C3%A1scara-de-led-que-reage-a-voz-a2c5fcd51cb2>