

Projeto de Extensão 2020

Relato do desenvolvimento do dispositivo Piscaledistância

Em 2020, o IFC Campus Blumenau realizou um projeto de extensão denominado **Dispositivos robóticos como ferramenta para estimular o interesse na área da programação**, no qual desenvolveu protótipos usando o arduino, socializando informações com docentes e discentes da escola básica municipal Lauro Muller. Este documento descreve o desenvolvimento de um protótipo nomeado **Piscaledistância** (Figura 1).

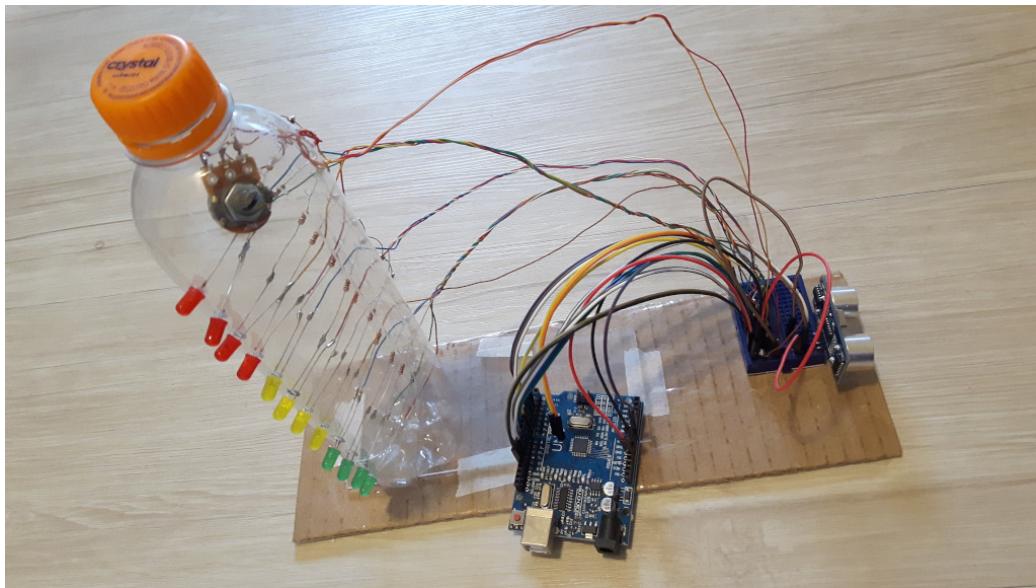


Figura 1: Protótipo 01: pistaledistancia.

O protótipo contém 12 leds e um potenciômetro ajustados em uma garrafa pet, e um sensor de distância fixado em um pequeno protoboard que conecta os leds, o potenciômetro e o sensor à placa arduino. Quando um objeto se aproxima ou afasta do sensor de distância, os leds acendem ou apagam. Este documento detalha a construção do protótipo, cujo funcionamento pode ser visto em <https://youtu.be/yipgbLpzbc0>.



2 TESTES PRELIMINARES

1 Concepção

O protótipo a ser construído deveria ser algo interessante mas de complexidade baixa ou média. Pensou-se então em um dispositivo que pudesse acender leds de acordo com a distância que um objeto estivesse do dispositivo. Os componentes envolvidos nesse equipamento são o **led**, que é o elemento que acende ou apaga, e um **sensor de distância**, para fornecer a informação de proximidade. Foi utilizado também um **potenciômetro**, que permite, por exemplo, alterar o funcionamento do sistema, e assim implementar mais de um modo de aproximação a ser considerado no sistema.

2 Testes preliminares

Antes de iniciar a construção do protótipo, é importante fazer testes para conhecer os componentes, bem como instalar o software de programação do arduino. Dessa forma vamos adquirindo confiança no manejo dos elementos, bem como ganhando ânimo para realizar ações mais elaboradas.

2.1 Piscar o led da placa

O primeiro teste realizado foi com a utilização do led que existe disponível na própria placa do arduino (Figura 2). A placa é conectada ao computador por meio de um cabo USB, e o led que é acionado está inserido na própria placa, não sendo necessário, neste momento, realizar conexões externas a outros componentes eletrônicos. Pode-se, assim, experimentar o processo mínimo para realizar a programação do arduino e verificar o funcionamento do sistema.



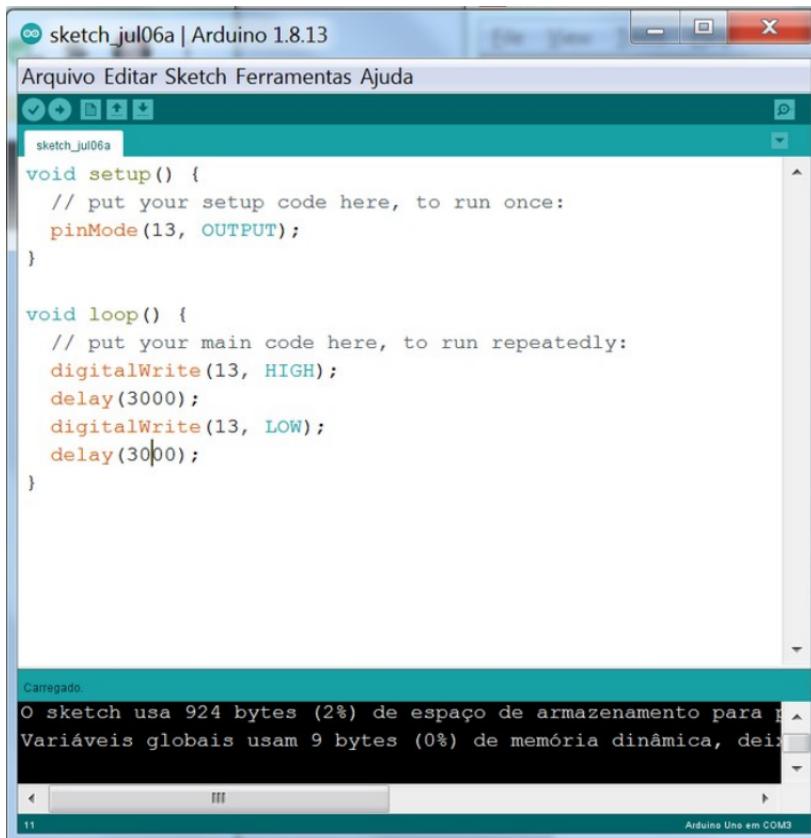
Figura 2: Piscar led do arduino



2.1 Piscar o led da placa

2 TESTES PRELIMINARES

O código que faz piscar o led da placa é pequeno e inicial (Figura 3). O software que realiza a programação do arduino pode ser obtido no site <https://www.arduino.cc/en/software>.



```

sketch_jul06a | Arduino 1.8.13
Arquivo Editar Sketch Ferramentas Ajuda
sketch_jul06a
void setup() {
    // put your setup code here, to run once:
    pinMode(13, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(13, HIGH);
    delay(3000);
    digitalWrite(13, LOW);
    delay(3000);
}

```

Carregado.
O sketch usa 924 bytes (2%) de espaço de armazenamento para p...
Variáveis globais usam 9 bytes (0%) de memória dinâmica, dei...

Figura 3: Código que faz o led da placa piscar

O programa possui uma parte de configuração que é executada apenas uma vez (*setup*). Nessa seção, o pino 13 do arduino é configurado para funcionar como saída de dados (*OUTPUT*), na qual pode ser informado um valor que liga ou desliga o led. Outra seção do código é executada de forma repetida (*loop*), cuja ação é enviar um sinal para ligar o led (*HIGH*), aguardar três segundos (*delay*), desligar o led (*LOW*) e aguardar novamente três segundos. Assim, obtém-se o efeito “piscar o led” da placa arduino.

2.2 Simulação

2 TESTES PRELIMINARES

2.2 Simulação

Para avançar no desenvolvimento do protótipo, o passo seguinte foi construir um programa maior que pudesse fazer uso dos componentes desejados: led, sensor de distância e potenciômetro. Existe um simulador gratuito de arduino na internet, no endereço <https://www.tinkercad.com/>. Foram criados alguns modelos para simular um led piscando, controlar potenciômetro, dentre outros. Por fim, foi criado um modelo que pisca diversos leds conforme a velocidade informada pelo potenciômetro (Figura 4).

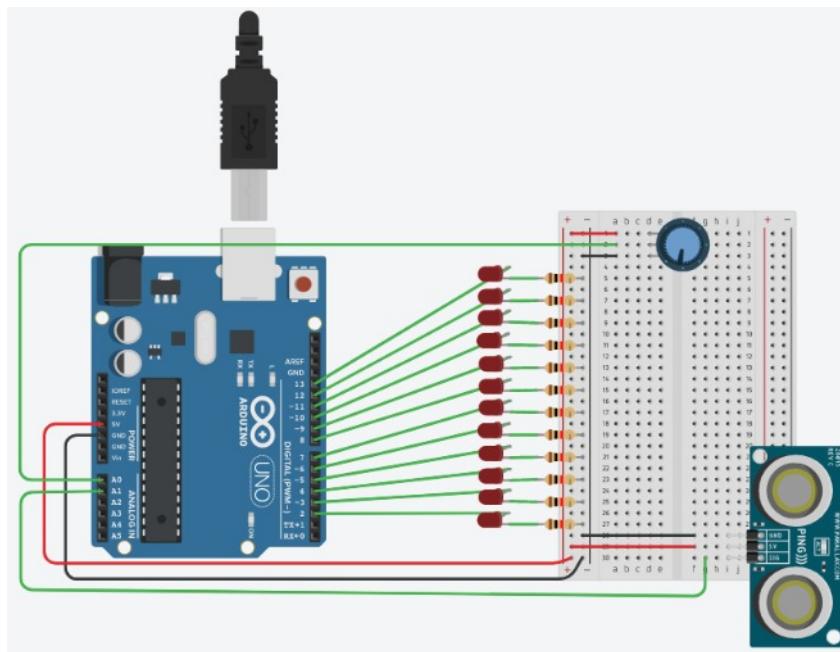


Figura 4: Simulação do protótipo: layout dos componentes

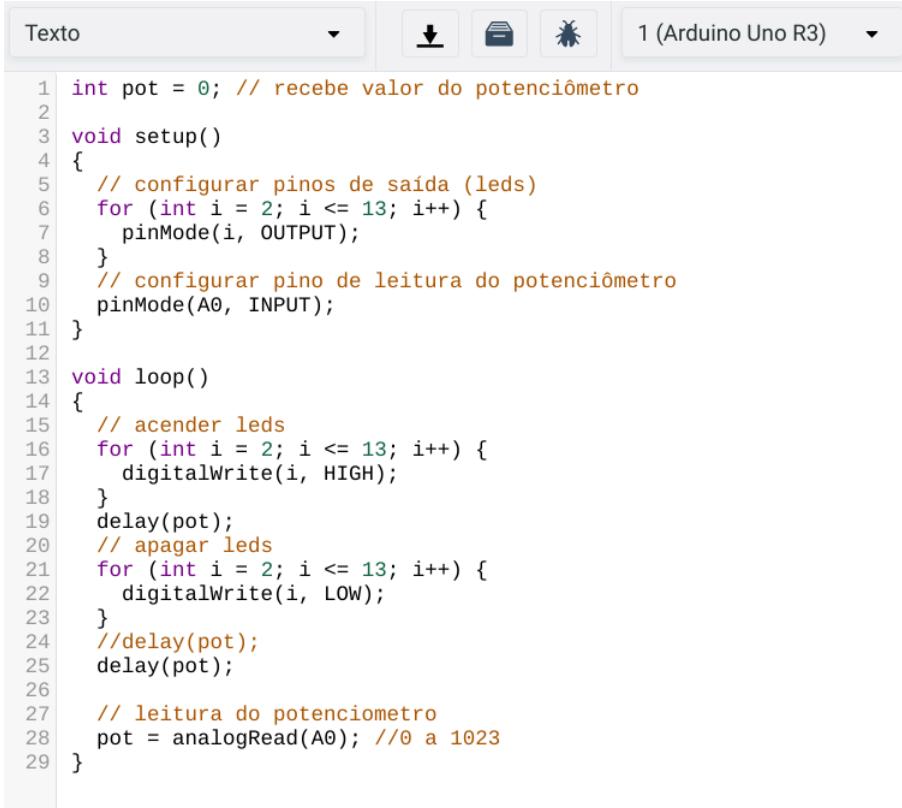
No modelo foram feitas conexões de leds, potenciômetro e sensor de distância, mas o sensor de distância não foi utilizado. Cada led teve uma extremidade conectada com o negativo (*ground*) e a outra extremidade conectada com uma porta digital do arduino. O potenciômetro foi conectado aos terminais positivo e negativo, enquanto o “fio do meio” foi conectado à porta analógica A0 do arduino.



2.2 Simulação

2 TESTES PRELIMINARES

O programa da simulação iniciou declarando uma variável que vai armazenar os valores do potenciômetro (Figura 5, linha 1). No código de configuração, que é executado apenas uma vez, os pinos de 2 a 13 são configurados como pinos de saída (linhas 6 a 8). Esses pinos são conectados aos leds, que serão acesos e apagados. O pino A0 é configurado como entrada (linha 10), pois nesse pino será realizada a leitura do valor do potenciômetro.



```

1 int pot = 0; // recebe valor do potenciômetro
2
3 void setup()
4 {
5     // configurar pinos de saída (leds)
6     for (int i = 2; i <= 13; i++) {
7         pinMode(i, OUTPUT);
8     }
9     // configurar pino de leitura do potenciômetro
10    pinMode(A0, INPUT);
11}
12
13 void loop()
14 {
15     // acender leds
16     for (int i = 2; i <= 13; i++) {
17         digitalWrite(i, HIGH);
18     }
19     delay(pot);
20     // apagar leds
21     for (int i = 2; i <= 13; i++) {
22         digitalWrite(i, LOW);
23     }
24     //delay(pot);
25     delay(pot);
26
27     // leitura do potenciometro
28     pot = analogRead(A0); //0 a 1023
29 }

```

Figura 5: Simulação do protótipo: código do programa

Na sequência de execução do programa, primeiramente todos os leds são acesos (linhas 16 a 18). A seguir, espera-se por um tempo cujo valor é definido pela variável que armazena o valor do potenciômetro (linha 19). Esse valor é zero na primeira execução do programa, mas será alterado em breve. Na sequência, apagam-se todos os leds (linhas 21 a 23) e espera-se novamente

3 MONTAGEM DO PROTÓTIPO

por um tempo definido em variável (linha 25). A seguir, é feita a leitura do potenciômetro (linha 28), quando então será atualizado o valor da variável que controla os tempos de espera, controlando assim a velocidade com a qual os leds irão piscar.

Após realizar o experimento básico fisicamente (piscagem do led do arduino) e experimentar simulações com outros componentes, estamos prontos para iniciar a montagem do protótipo.

3 Montagem do protótipo

A primeira ação de construção do protótipo foi a instalação de leds de forma organizada em uma garrafa pet (Figura 6). A ideia foi demonstrar que elementos simples (no caso, a garrafa pet) podem ser utilizados para montar protótipos eletrônicos. Cada led foi soldado a um resistor, de modo que foram instalados na garrafa 12 leds.

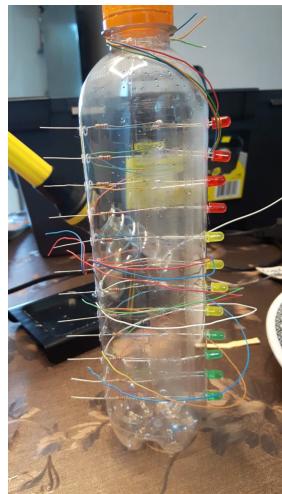


Figura 6: Montagem de leds em uma garrafa pet

O próximo passo na construção do protótipo foi a instalação do potenciômetro (Figura 7). O componente ficou localizado próximo ao topo da garrafa, com as conexões direcionadas para o exterior da garrafa por meio de

3 MONTAGEM DO PROTÓTIPO

fios soldados ao potenciômetro. Os fios dos leds e do potenciômetro foram reunidos em grupos, permitindo a diferenciação por conjuntos de cores, de forma que cada grupo de fios possui cores diferenciáveis. Os fios foram conectados a uma mini-protoboard auxiliar, organizando as sequências de leds. O sensor de distância também foi conectado no mini-protoboard.

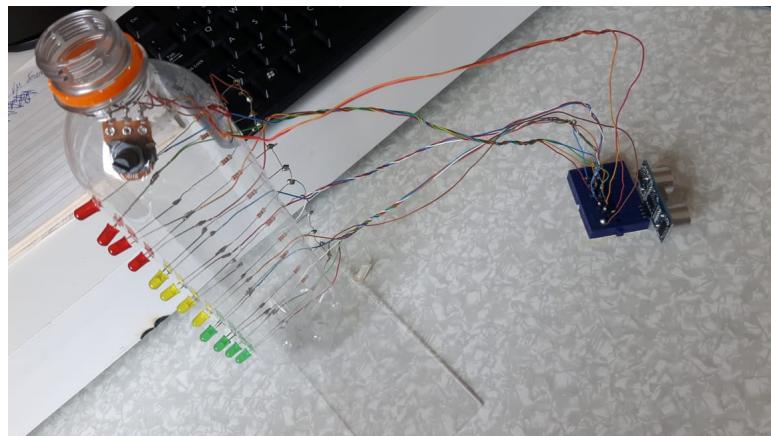


Figura 7: Instalação do potenciômetro e conexões à mini-protoboard auxiliar

O protótipo e foi então conectado ao arduino, nas devidas portas (Figura 8).

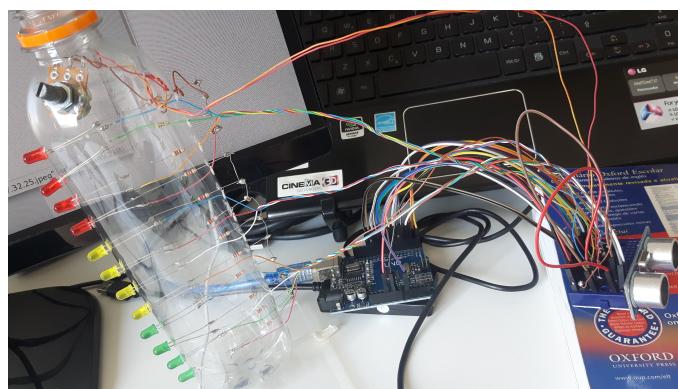


Figura 8: Conexão do protótipo ao arduino



4 PROGRAMAÇÃO

O funcionamento do protótipo contou com duas funções principais:

- **Lógica 1:** quando um objeto se aproxima do protótipo, as luzes permanecem apagadas até que o objeto alcance uma distância mínima. A partir dessa distância mínima, qualquer distância mais próxima do protótipo faz com que todos os leds se acendam.
- **Lógica 2:** todas as luzes ficam acesas, mas na medida em que um objeto se aproxima do protótipo, as luzes vão se apagando gradativamente.

A seleção da lógica que está sendo utilizada se dá por meio do potenciômetro. Considerando que não há objeto em frente (próximo) do sensor de distância, quando o potenciômetro é girado para um lado, a primeira lógica é utilizada e todos os leds são apagados. Quando o potenciômetro é girado para o outro lado, a segunda lógica é acionada e todos os leds são acesos.

4 Programação

O programa utilizado no protótipo está disponível no repositório público <https://github.com/hvescovi/arduino/tree/main/2020>, dentro da pasta denominada `07-dois-algoritmos-selecao-potenciometro`. Dentro pequenos códigos disponíveis de testes, está o código completo que realiza as duas funcionalidades de aproximação por limite e aproximação gradual (Código 1). O programa inicia com a declaração de variáveis que identificam os pinos utilizados no sensor de distância para conectar os terminais trigger (linha 1) e echo (linha 2). Uma variável é utilizada para calcular o tamanho da onda emitida pelo sensor de distância (linha 4) e outras duas variáveis irão armazenar a distância percorrida pela onda, em polegadas (linha 5) e centímetros (linha 6). Em uma variável chamada `maxDist` (linha 9) será armazenado o valor máximo que corresponde ao acendimento de todos os leds. Esse valor foi definido como vinte (20), de forma que o salto seja calculado igual a dois (2) em uma variável chamada `salto` (linha 13). Isso significa que a cada variação de duas unidades na distância do sensor um novo led será aceso ou apagado. Por fim, declara-se uma variável para armazenar o valor que será lido do potenciômetro (linha 16).



4 PROGRAMAÇÃO

```

1 int pingPin = 2; // mapeamento do pino trigger
2 int echoPin = 3; // mapeamento do pino echo
3
4 float duration; // tamanho da onda
5 float inches; // polegadas
6 float cm; // centímetros
7
8 // valor escolhido observando variação no sensor de distância em
   → centímetros
9 int maxDist = 20;
10
11 // são 10 leds a acender
12 // divide um intervalo de máximo de polegadas em 10 partes, em
   → centímetros
13 int salto = maxDist / 10;
14
15 // valor a ser lido do potenciômetro
16 int potenciometro = 0;

```

Código 1: Programa do protótipo Piscaledistância - parte 1

Fonte: <https://github.com/hvescovi/arduino/tree/main/2020/>,
 pasta: 07-dois-algoritmos-selecao-potenciometro

A inicialização do arduino contém um código que é executado apenas uma vez (Código 2). Primeiramente, se inicializa a porta de comunicação serial (linha 20), para que possam ser exibidas mensagens durante a execução do programa. Isso é útil para visualizar valores de variáveis durante a execução do programa. Na sequência, as portas digitais são inicializadas, endereçando os 10 leds que serão utilizados pelo programa (linhas 23 a 25). A garrafa de leds possui 12 leds, mas dois deles não serão utilizados. O motivo é que quando se realiza comunicação pela porta serial, as portas digitais 0 e 1 não devem ser utilizadas. As portas 2 e 3 são utilizadas pelo sensor de distância, restando assim as portas 4 a 13 para os leds. Por fim, duas portas utilizadas pelo sensor são configuradas: a porta conectada ao *trigger* (linha 28) e a porta conectada ao *echo* (linha 29).



4 PROGRAMAÇÃO

```

18 void setup() { // código executado somente uma vez
19   // inicialização do canal serial de comunicação
20   Serial.begin(9600);
21
22   // configurar pinos de saída (leds)
23   for (int i = 4; i <= 13; i++) { // 10 leds
24     pinMode(i, OUTPUT);
25   }
26
27   // configurar pinos do sensor de distância
28   pinMode(pingPin, OUTPUT); // pino trigger (envia onda)
29   pinMode(echoPin, INPUT); // pino echo (recebe informação)
30 }
31
32 long microsecondsToInches(long microseconds) {
33   return microseconds / 74 / 2;
34 }
```

Código 2: Programa do protótipo Piscaledistância - parte 2

Fonte: <https://github.com/hvescovi/arduino/tree/main/2020/>,
 pasta: 07-dois-algoritmos-selecao-potenciometro

Duas funções são criadas para facilitar o uso do sensor de distância (Código 3). A primeira função converte tempo para polegadas, enquanto a segunda função converte uma informação temporal para centímetros.

```

32 long microsecondsToInches(long microseconds) {
33   return microseconds / 74 / 2;
34 }
35 long microsecondsToCentimeters(long microseconds) {
36   return microseconds / 29 / 2;
37 }
```

Código 3: Programa do protótipo Piscaledistância - parte 3

Fonte: <https://github.com/hvescovi/arduino/tree/main/2020/>,
 pasta: 07-dois-algoritmos-selecao-potenciometro



4 PROGRAMAÇÃO

Uma função específica foi definida para realizar a manipulação de leds na garrafa de leds (Código 4). A função `acionar_leds` liga os leds até o n-ésimo led. Por exemplo, se for realizada a chamada `acionar_leds5`, serão ligados os leds 1, 2, 3, 4 e 5, enquanto os demais leds (6 a 10) serão desligados. A função inicia apagando todos os leds (linhas 42 a 44) e depois acendendo somente aqueles que se encontram no intervalo de 4 até o valor 4 somado do número específico, menos 1 (linhas 46 a 48). Para um exemplo de ligar os 5 primeiros leds, a variação desse intervalo seria entre 4 e $4+5-1$, ou seja, de 4 a 8. Isso significa ligar os leds que estão conectados nas portas 4, 5, 6, 7 e 8.

```

39 // aciona os leds até o n-ésimo led
40 void acionar_leds(int n) {
41     // apaga todos os leds
42     for (int i = 4; i <= 13; i++) {
43         digitalWrite(i, LOW);
44     }
45     // acende os leds até o desejado
46     for (int i = 4; i <= 4+n-1; i++) {
47         digitalWrite(i, HIGH);
48     }
49 }
```

Código 4: Programa do protótipo Piscaledistância - parte 4

Fonte: <https://github.com/hvescovi/arduino/tree/main/2020/>,
 pasta: 07-dois-algoritmos-selecao-potenciometro

A ação de controle dos leds é executada de acordo com o algoritmo que está sendo executado, cuja seleção é feita de acordo com o potenciômetro. Ambas as lógicas consideram a distância com a qual um objeto se encontra do protótipo, sendo que na primeira lógica existe apenas uma condição de proximidade limite. Para a primeira lógica foi desenvolvida uma função que verifica se o objeto está “longe” (Código 5), sendo que esse “longe” foi definido como a medida de dez (10) centímetros, mensurada pelo sensor de distância (linha 52). Quando um objeto se encontra “longe”, todos os leds são apagados (linha 53). Caso contrário, o objeto ultrapassou o limite estabelecido e está “perto”. Sendo assim, todos os leds são acesos para sinalizar esse cenário (linha 55).



4 PROGRAMAÇÃO

```

51 void algoritmo1(int cm) {
52     if (cm >= 10) { // está longe?
53         acionar_leds(0); // apaga os leds
54     } else { // senão (está próximo)...
55         acionar_leds(10); // acende os leds!
56     }
57 }
```

Código 5: Programa do protótipo Piscaledistância - parte 5

Fonte: <https://github.com/hvescovi/arduino/tree/main/2020/>,
pasta: 07-dois-algoritmos-selecao-potenciometro

Uma segunda função foi criada para armazenar a lógica que acende os leds gradualmente conforme a distância com a qual o objeto se encontra do protótipo (Código 6). A variação de valores medidos pelo sensor de distância foi observada, ocorrendo em uma faixa aproximadamente igual a 1 e 20, sendo o valor 1 difícil de se alcançar, e valores maiores do que 20 realizam um incremento brusco para números maiores. Dessa forma, com o salto calculado no começo do programa (igual a 2), foram verificados os intervalos entre 2 e 20 para controlar o acendimento de leds. A primeira condição verificada é se o objeto se encontra a uma distância maior do que 20 (linha 61). Em caso positivo, o objeto está “longe” e todos os leds permanecem acesos. Caso a distância percebida pelo sensor seja menor do que 20, verifica-se, na sequência, se o valor permanece maior do que 18. Em caso positivo, um led é desligado e os outros 9 permanecem ligados. Seguindo-se essa lógica, as condições subsequentes são verificadas, até o caso em que a distância é menor do que 2 (caso raro, devido à sensibilidade do sensor), e todos os leds são desligados (linha 82). Esse algoritmo da lógica 2 pode ser melhorado, substituindo as diversas condições por uma condição dentro de um laço de repetição. Deixamos a cargo do leitor a realização deste exercício, cuja solução deve ser disposta futuramente no repositório de código indicado neste material.

```

59 void algoritmo2(int cm) {
60     // verifica distância proporcional ao máximo
61     if (cm >= salto * 10) { // > 20?
62         acionar_leds(10);
```



4 PROGRAMAÇÃO

```

63 } else if (cm >= salto * 9) { // > 18?
64   acionar_leds(9);
65 } else if (cm >= salto * 8) { // > 16?
66   acionar_leds(8);
67 } else if (cm >= salto * 7) {
68   acionar_leds(7);
69 } else if (cm >= salto * 6) {
70   acionar_leds(6);
71 } else if (cm >= salto * 5) {
72   acionar_leds(5);
73 } else if (cm >= salto * 4) {
74   acionar_leds(4);
75 } else if (cm >= salto * 3) {
76   acionar_leds(3);
77 } else if (cm >= salto * 2) { // > 4?
78   acionar_leds(2); // acende 2 leds
79 } else if (cm >= salto) { // > 2?
80   acionar_leds(1); // acende 1 led
81 } else {
82   acionar_leds(0); // não acende nenhum led
83 }
84 }
```

Código 6: Programa do protótipo Piscadedistância - parte 6

Fonte: <https://github.com/hvescovi/arduino/tree/main/2020/>,
pasta: 07-dois-algoritmos-selecao-potenciometro

Finalmente, o trecho de código que é executado repetidamente realiza o controle principal das ações do protótipo (Código 7). O programa começa emitindo sinalizações para o sensor de distância, a fim de emitir uma onda ultrasônica (linhas 88 a 92). A largura do pulso da onda pode ser medida com uso da função `pulseIn` disponível na linguagem de programação do arduino (linha 93). A seguir, são calculadas em polegadas (linha 95) e centímetros (linha 96) as distâncias de um possível objeto que se encontra em frente ao sensor de distância. Essas informações são exibidas no terminal serial (linhas 98 a 102), que pode ser observado quando se utiliza o programa que carrega



4 PROGRAMAÇÃO

as instruções no arduino. Um pequeno tempo de espera é solicitado (linha 104) e então a informação do potenciômetro é lida (linha 108), valor esse expresso em termos números, em um intervalo de 0 (potenciômetro totalmente girado para um lado) a 1023 (potenciômetro totalmente girado para o outro lado). Dependendo a posição do potenciômetro (linha 112), a lógica de aproximação limite é utilizada (linha 113) ou a lógica de aproximação gradual é considerada (linha 115).

```
86 void loop() { // código executado repetidamente
87     // emissão de sinais do sensor de distância
88     digitalWrite(pingPin, LOW);
89     delayMicroseconds(2);
90     digitalWrite(pingPin, HIGH);
91     delayMicroseconds(10);
92     digitalWrite(pingPin, LOW);
93     duration = pulseIn(echoPin, HIGH);
94     // cálculos de distância
95     inches = microsecondsToInches(duration);
96     cm = microsecondsToCentimeters(duration);
97     // exibição de informações
98     Serial.print(inches);
99     Serial.print("in,");
100    Serial.print(cm);
101    Serial.print("cm");
102    Serial.println();
103
104    delay(100);
105
106    // leitura do valor do potenciômetro
107    // é lido um valor de 0 a 1023
108    potenciometro = analogRead(A0);
109
110    // verificação da lógica
111    // considera-se aproximadamente o meio do potenciômetro
112    if (potenciometro <= 511) {
113        algoritmo1(cm);
```



4 PROGRAMAÇÃO

```
114 } else {  
115     algoritmo2(cm);  
116 }  
117 }
```

Código 7: Programa do protótipo Piscaledistância - parte 7

Fonte: <https://github.com/hvescovi/arduino/tree/main/2020/>,
pasta: 07-dois-algoritmos-selecao-potenciometro

Em uma etapa final da confecção do protótipo, foi colocado um recorte de papelão por baixo dos componentes, a fim de melhor fixar a garrafa de leds e o mini-protoboard (Figura 1).