

# Jogo “Descobre o LED”

## Introdução

Este trabalho é realizado no âmbito da disciplina de Tecnologias dos Computadores e tem como objetivo desenvolver um programa que implemente o jogo intitulado “Descobre o LED”.

Como é falado no enunciado do trabalho e passo a citar “O objetivo é conseguir acender todos os leds com o menor número de cliques.”

## Explicação de como funciona o programa por mim escrito

Para começar foi definido um array bidimensional onde dentro do primeiro do primeiro elemento desse array foi definido os “pins” de cada LED(3 a 10) e sendo que o segundo elemento serve para definir os estados de cada LED(definidos inicialmente como “LOW”).

De seguida foram definidos os “pins” do botão que vai acender o LED quando o descobrir(Botão 1) e também do botão de reset(Botão 2).

Por último foi definida uma variável que contará o número de cliques dados no Botão 1 bem como duas variáveis do tipo “bool” para o estado corrente do Botão 1 e outra para o estado anterior do Botão 1.

## Void Setup:

Dentro da função “Void Setup” foi definido os “pinMode’s” dos LED’s como “OUTPUT”(usando um ciclo “for”) para que estes sejam posteriormente acesos. Foi definido também os “pinModes’s” do Botão 1 e Botão 2 como “INPUT\_PULLUP”.

Foi ainda inicializado o “Serial monitor” usando “Serial.begin(9600)” para no final do jogo ser apresentado o números de cliques dados pelo Botão 1.

## Void Loop:

Passando agora finalmente para a função “Void Loop”. Começou-se por criar um ciclo “for” para acender e apagar os LED’s sequencialmente. Dentro desse mesmo ciclo além de definir o estado do LED para HIGH(acender) ou LOW(apagar) foram introduzidos “delays” entre a ação de acender e apagar o LED.

Esses “delays” não se limitaram a ser um simples “delay(tempo definido)”. Criei um ciclo “for”(que ia de 0 a 150, que foi o tempo que defini para mudança de estado entre LED’s, em milissegundos). Dentro desse mesmo ciclo passei a colocar a ação de acender o LED(já antes referida), depois as condições que diriam que se carregasse no Botão 1 enquanto um LED

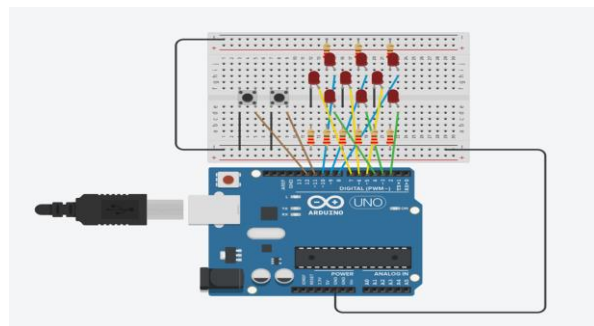
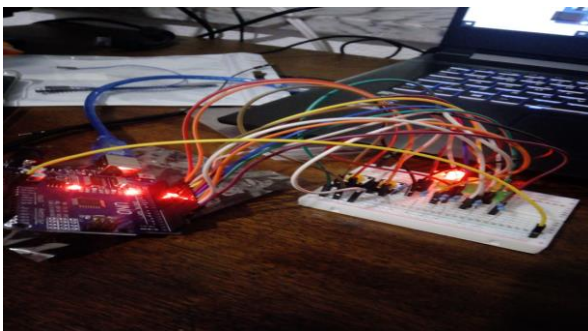
estava aceso ele se manteria aceso (e contaria um clique dado nesse botão) e se não carregasse o LED apagaria e passava para o próximo. Dentro destas duas condições coloquei “delay’s(1)” pois visto que o ciclo era de 0 a 150, faria com que entre cada ação houve-se um espaço de 150 milissegundos.

Provavelmente estará-se a perguntar o porquê de todo este drama em relação ao “delay”. A resposta é simples, apenas o implementei desta forma para ser possível fazer alterações enquanto o tempo entre cada ação(acender e apagar o LED), alterações essas referentes a deixar o LED permanentemente aceso quando carregava no botão.

Ainda dentro do ciclo “for” de acender e apagar os LED’s sequencialmente foi implementado o código do Botão 2, o qual daria reset ao jogo, o qual é simples. Definir “status2” como o estado lido no Botão 2 e uma condição para quando o estado do botão for “LOW”(Isto porque o Botão 2 foi definido como “INPUT\_PULLUP”) o estado dos LED’s passaria a ser “LOW” de novo e “escreveria” em todos os LED’s o valor “LOW”(utilizando um ciclo “for”). Além disto o contador de cliques dado no Botão 1 voltaria a 0.

(Acontece que o botão de reset não está totalmente como o pedido e o Jogo em vez de dar “total” reset e começar a acender os LED’s desde o primeiro, acende apartir do último que esteve aceso). Encerra-se aqui o ciclo “for” inicialmente aberto.

Passamos então para a última parte do código que é a “Funcionalidade vitória”. O código aqui também é bastante simples. Criei uma condição para que se todos os LED’s estiverem ao mesmo tempo acesos apresentaria uma mensagem no “Serial monitor” de quantos cliques se deram no Botão 1 e os LED’s piscariam durante 20 segundos(sempr sempre ao mesmo tempo), para isso voltei a criar um novo ciclo “for” que vai de 0 a 20 sendo que dentro dele “escreveria” em todos os LED’s o valor “HIGH” dava um delay de 500 milissegundos e “escreveria” o valor “LOW” dando outro delay de 500 milissegundos(para passar 1 segundo entre cada piscar). Além disso voltei a definir o estado de todos os LED’s como “LOW” e reiniciar o contador do número de cliques dados no Botão 1 fazendo com que o Jogo reiniciasse passados os 20 segundos(tal como pedido).



#### Referências Usadas:

<https://www.arduino.cc/reference/en/language/variables/data-types/array/>