



UNIVERSIDADE ESTADUAL DA PARAÍBA  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
BACHARELADO EM CIÊNCIAS  
DA COMPUTAÇÃO

LUCAS DE LUCENA SIQUEIRA

DISCIPLINA: LABORATÓRIO DE ORGANIZAÇÃO E ARQUITETURA DE  
COMPUTADORES

EXPERIMENTO 05

**ARDUINO - TINKERCAD**

CAMPINA GRANDE - PB

2021

## SUMÁRIO

1. RESUMO.....	3
2. INTRODUÇÃO.....	3
3. MATERIAL E MÉTODOS.....	3
3.1. OBJETIVO.....	3
3.2. SOFTWARE NECESSÁRIO.....	3
3.3. HARDWARE NECESSÁRIO.....	3
3.4. PROCEDIMENTOS.....	3
3.4.1.MONTAGEM DO HARDWARE.....	3
3.4.2. LISTAGEM DO PROGRAMA EM C++.....	4
3.4.3. FUNCIONAMENTO DO PROGRAMA.....	6
3.4.4. DESCRIÇÃO E APLICAÇÕES DO PROJETO.....	6
4. RESULTADOS E DISCUSSÃO.....	7
5. CONCLUSÕES.....	7
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	7

## 1. RESUMO

Neste trabalho em questão é possível observar alguns conceitos básicos e importantes sobre a programação do Arduino em C++ a partir do simulador Tinkercad da Autodesk. Por fim, foram feitas as atividades propostas pelo professor e expostas neste relatório para a prática e conhecimento do microcontrolador Arduino bem como da proposição de comandos que poderão acionar buzinas, leds, motores e botões.

## 2. INTRODUÇÃO

O presente trabalho tem como objetivo introduzir o estudo do microcontrolador Arduino a partir do uso do simulador Tinkercad da Autodesk. Tendo o C++ como linguagem dominante na programação do referente microcontrolador, se faz necessário também o estudo da mesma a partir de consultas em seu *Datasheet*, que é nada mais do que uma folha com dados e especificações técnicas e de desempenho do produto levado em consideração, que no nosso caso é o microcontrolador Arduino.

## 3. MATERIAL E MÉTODOS

### 3.1. OBJETIVO

O experimento em questão tem como objetivo induzir o estudo e a prática da programação do microcontrolador Arduino a partir do Tinkercad da Autodesk com o uso da linguagem C++. Tornando possível estudar o acionamento de LED's, motores, buzinas e botões.

### 3.2. HARDWARE NECESSÁRIO

- Computador com sistema operacional com browser de internet

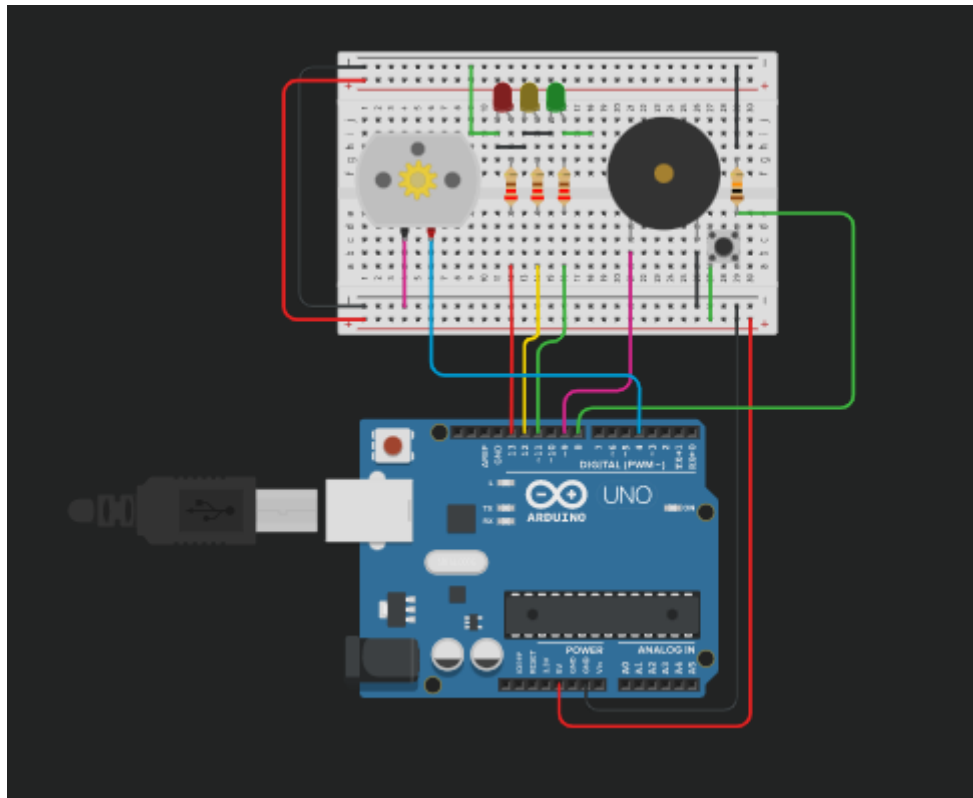
### 3.3. SOFTWARE NECESSÁRIO

- Simulador Tinkercad.

### 3.4. PROCEDIMENTOS

#### 3.4.1. MONTAGEM DO HARDWARE

Na figura abaixo estão as conexões realizadas no Arduino Uno R3, Led's, Buzina Piezo, Motor CC, Resistores e no Botão:



### 3.4.2. LISTAGEM DO PROGRAMA EM C++

Um programa feito para C++ é subdividido em dois grupos principais, o setup e o loop, o setup é responsável por designar as funções lógicas dos pinos, sejam eles de entrada ou de saída. Já o loop é a parte do programa responsável por realizar as ações desejadas, sendo essa uma função que fica se repetindo infinitamente.

- void setup() {}

Um primeiro ponto importante a ser citado diz respeito ao funcionamento de da função “pinMode” e dos parâmetros que são dados para a mesma. A função “pinMode” tem por objetivo definir se o pino que será passado como primeiro parâmetro será de saída ou de entrada, a partir da definição como “INPUT” ou “OUTPUT”, para entrada e saída respectivamente, como segundo parâmetro da função.

Segue o corpo da função setup():

```

3  int botao = 0;
4
5  void setup()
6  {
7      pinMode(8, INPUT);
8      pinMode(11, OUTPUT);
9      pinMode(12, OUTPUT);
10     pinMode(13, OUTPUT);
11     pinMode(9, OUTPUT);
12     pinMode(4, OUTPUT);
13     pinMode(9, OUTPUT);
14 }
15

```

- void loop()

Nesse trecho existem algumas funções a mais a serem explicadas. A mais importante diz respeito ao “digitalWrite” e ao “digitalRead”, que vão, respectivamente, definir o sinal elétrico como “HIGH” ou “LOW”, além dessa função, há o “delay” que recebe como parâmetro o tempo em milissegundos de espera antes de seguir para a próxima linha de código e por fim, temos o “tone”, que vai receber três parâmetros, o primeiro referente ao pino o Arduino e os outros dois referentes à frequência que será utilizada na buzina Piezo. Sendo Assim, segue o trecho do código referente à função loop():

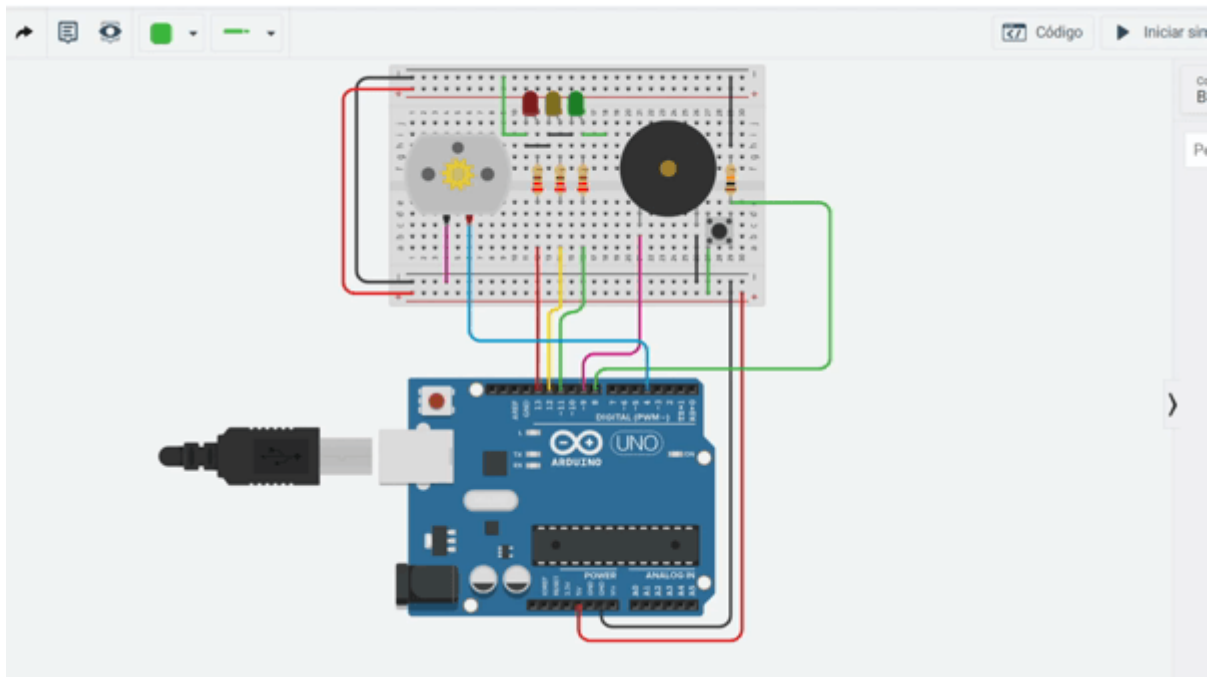
```

16 void loop()
17 {
18     if (digitalRead(8) == HIGH) {
19         digitalWrite(11, LOW);
20         digitalWrite(12, HIGH);
21         delay(3000); // Wait for 3000 millisecond(s)
22         digitalWrite(12, LOW);
23         digitalWrite(13, HIGH);
24         tone(9, 57079043007393, 5000); // play tone 500
25         digitalWrite(4, HIGH);
26         delay(5000); // Wait for 5000 millisecond(s)
27         digitalWrite(13, LOW);
28         digitalWrite(4, LOW);
29         digitalWrite(9, LOW);
30     }
31
32     digitalWrite(11, HIGH);
33 }

```

### 3.4.3. FUNCIONAMENTO DO PROGRAMA

Aqui estará presente todo o funcionamento do programa referente ao código já apresentado acima.



No primeiro momento do programa o led verde ficará sempre aceso até que a ação de apertar o botão seja realizada. Quando o botão é pressionado o led verde irá desligar e o amarelo ficará aceso durante 5 segundos, logo após os 5 segundos o led amarelo apaga e o vermelho irá ficar aceso por 5 segundos além de que o motor e a buzina irão ser ativados também, logo após esses 5 segundos com todos os componentes citados anteriormente em funcionamento, o programa irá voltar para o início novamente, mantendo o led aceso até que o botão seja pressionado novamente.

### 3.4.4. DESCRIÇÃO E APLICAÇÕES DO PROJETO

O projeto teve como intuito pôr em prática a utilização de conceitos e artifícios disponíveis do Arduino, como por exemplo aplicar condicionais e aplicar a utilização de botões, leds, motores, buzinas e resistores.

Esse circuito assim como outros possíveis circuitos a serem montados podem ter várias utilizações, como por exemplo programar um sensor de temperatura ou até um semáforo, ambos utilizando recursos sonoros, físicos ou visíveis.

#### 4. RESULTADOS E DISCUSSÃO

Após todo o experimento foi possível compreender algumas peculiaridades do arduino. Após a execução do código listado foi possível compreender a relação entre as operações realizadas, e as ações que eram expostas no simulador.

#### 5. CONCLUSÕES

Foi possível concluir que a linguagem C++ em atuação ao microcontrolador Arduino pode dar uma vasta biblioteca de opções e atuações diante de suas portas, assim como demonstrado no decorrer do relatório.

#### 6. REFERÊNCIAS BIBLIOGRÁFICAS

TUTORIALS | Arduino. [S. l.], 9 out. 2021. Disponível em:  
<https://www.arduino.cc/en/Tutorial/HomePage>. Acesso em: 9 out. 2021.