



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA  
ENGENHARIA MECATRÔNICA  
TÓPICOS ESPECIAIS EM SISTEMAS EMBARCADOS**

**RELATÓRIO DE IMPLEMENTAÇÃO - ACIONANDO UM MOTOR  
VIA BLYNK**

**ATYSON JAIME DE SOUSA MARTINS**

**Natal-RN  
MAIO/2021**

ATYSON JAIME DE SOUSA MARTINS

## **RELATÓRIO DE IMPLEMENTAÇÃO - ACIONANDO UM MOTOR VIA BLYNK**

Relatório apresentado à disciplina de tópicos especiais em sistemas embarcados, correspondente à avaliação da 3ª unidade do semestre 2020.2 do curso de Engenharia Mecatrônica da Universidade Federal do Rio Grande do Norte, sob orientação do **Prof. Dr. Victor Araújo Ferraz.**

Professor: Victor Araújo Ferraz.

Natal-RN  
MAIO/2021

## Lista de Figuras

1	Motores de naves espaciais . . . . .	5
2	Placa de Desenvolvimento Esp-32 . . . . .	6
3	Blynk logo . . . . .	7
4	Arquitetura Blynk . . . . .	7
5	Projeto no Blynk App . . . . .	8
6	Modelo montado em protoboard . . . . .	11
7	Tela inicial do aplicação . . . . .	12
8	Tela modo ligado do aplicação . . . . .	13
9	Circuito com motor ligado . . . . .	14
10	Tela modo desligado da aplicação . . . . .	15
11	Circuito com motor desligado . . . . .	16

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>5</b>
<b>2</b>	<b>DESENVOLVIMENTO</b>	<b>6</b>
2.1	ESP32-wroom-32 . . . . .	6
2.1.1	Wi-Fi . . . . .	6
2.2	Blynk . . . . .	7
2.2.1	Blynk App . . . . .	8
2.2.2	Blynk Server . . . . .	9
2.2.3	Blynk Libraries . . . . .	9
<b>3</b>	<b>RESULTADOS</b>	<b>11</b>
<b>4</b>	<b>CONCLUSÃO</b>	<b>17</b>

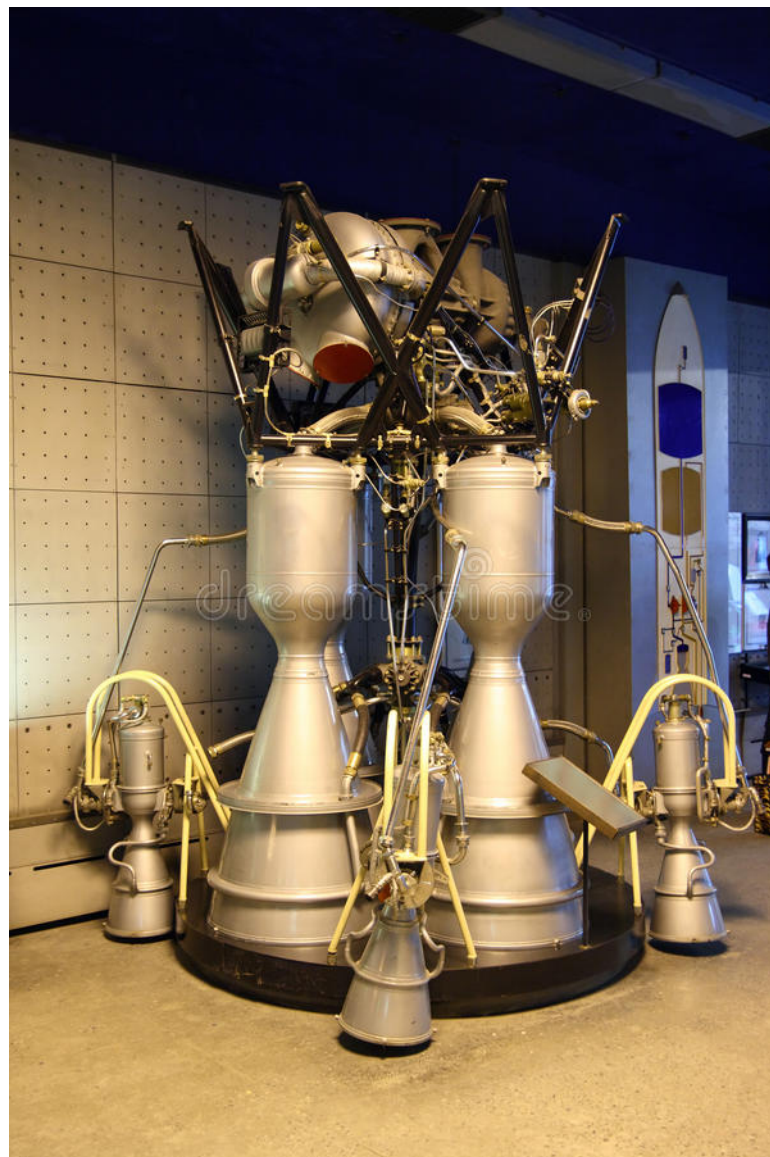
# 1 INTRODUÇÃO

O acionamento de motores ou turbinas a distancia são comumente utilizados em grandes aplicações como: motores de naves, turbinas de resfriamento em usinas hidrelétricas e nucleares 1. Desse modo, tornando seguro para os operadores trabalharem, sem a necessidade de está ao lado desses motores.

O uso de dispositivos controláveis por software como microcontroladores, auxiliam no processo de implementação desses tipos de controladores, possibilitando a implementação de centrais a distancia.

O presente trabalho tem por objetivo exibir a metodologia de projeto para desenvolvimento do acionamento de um motor via blynk.

Figura 1: Motores de naves espaciais



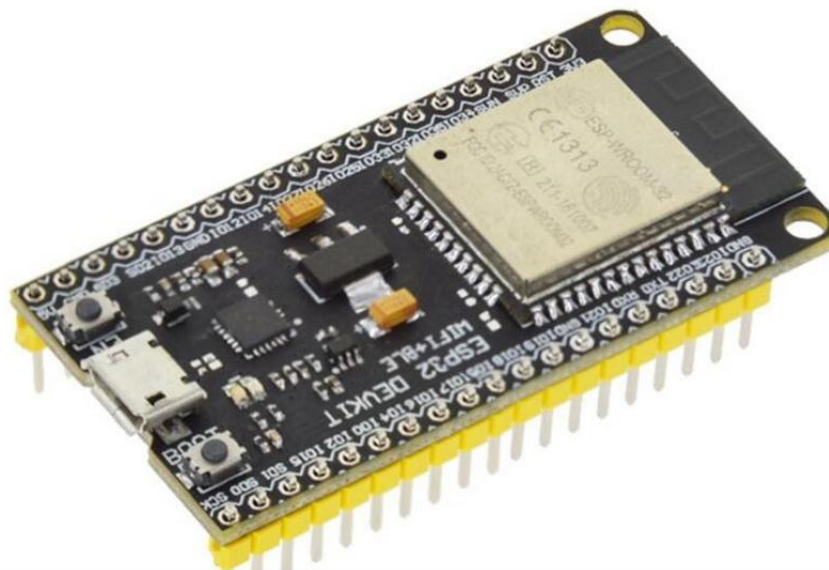
Fonte: Imagem disponibilizada no google.

## 2 DESENVOLVIMENTO

### 2.1 ESP32-wroom-32

ESP32 ou Esp-wroom-32 é uma placa de desenvolvimento criada pela empresa (*Espressif Systems*), possui como sua antecessora o ESP8266. Este pequeno componente é bastante difundido em aplicações de automações no geral, pois em sua placa além de apresentar módulo de comunicação Wi-Fi, apresenta um sistema com processador Dual Core, Bluetooth híbrido e múltiplos sensores embutidos, tornando a construção de sistemas mais fácil e simplificado. Para esse projeto, utilizamos o modulo de comunicação wi-fi, a posteriori será melhor explicado sobre esse modulo. Outrossim, para conseguir desenvolver e programar o esp-32 foi usada a IDE do Arduíno.

Figura 2: Placa de Desenvolvimento Esp-32



---

Fonte: Imagem disponibilizada no google

#### 2.1.1 Wi-Fi

Algumas características desse módulo são as seguintes:

- IEEE 802.11 b/g/n (2.4 GHz até 150 Mbps);
- WPA/WPA2/WPA2-Enterprise e WPS;
- AMPDU, HT40 e QoS;
- Suporta vários modos (Infrastructure Station, SoftAP, Station+AP, Promíscuo (Sniffer));

Como a IDE utilizada foi a do Arduino, precisamos importar a biblioteca wifi como as funções que iremos utilizar. Desse modo, iremos configurar o código da seguinte maneira:

```
#include <WiFi.h>

// Wifi config
char ssid[] = "-----"; // Nome da sua rede
char pass[] = "-----"; // Senha da sua rede
```

## 2.2 Blynk

Esse serviço é baseado em um aplicativo personalizável que permite controlar remotamente um hardware programável, bem como reportar dados do hardware ao aplicativo. Dessa forma, é possível construir interfaces gráficas de controle de forma rápida e intuitiva.

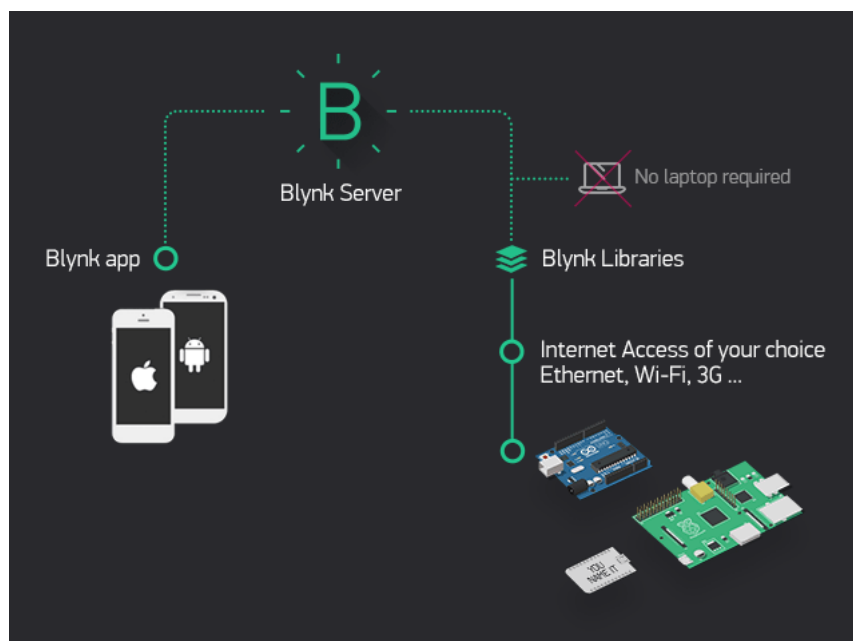
Figura 3: Blynk logo



Fonte: Site oficial da blynk.

O blynk é dividido em três partes: Blynk App, Blynk Server e Blynk Library conforme a figura 4.

Figura 4: Arquitetura Blynk

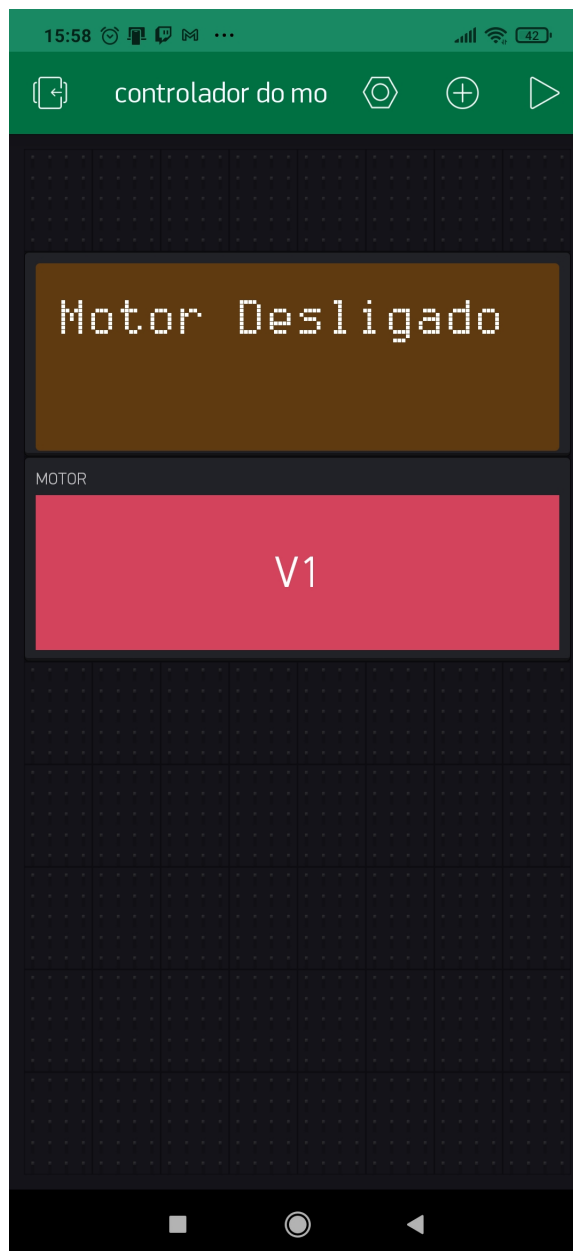


Fonte: Imagem disponibilizada no google.

### 2.2.1 Blynk App

é um aplicativo disponível para Android e iOS, no qual, permite o usuário criar aplicativos para interagir com o hardware através de (*widgets*), tem como sua moeda a energia, onde dependendo da funcionalidade escolhida, consome um valor de energia. Para esse dois wid-gets foram utilizados, um LCD Display para mostrar o status do motor e um Botão para ligar o motor. Sendo assim, ficando da seguinte forma no aplicativo:

Figura 5: Projeto no Blynk App



Fonte: Autoria Própria



### 2.2.2 Blynk Server

Toda comunicação entre o aplicativo e o hardware do usuário se dá através da (*cloud Blynk*). O servidor é responsável por transmitir os dados ao hardware, armazenar estados do aplicativo e do hardware e também armazenar dados de sensores lidos pelo hardware mesmo se o aplicativo estiver fechado.

### 2.2.3 Blynk Libraries

Ao lado do hardware temos as bibliotecas Blynk para diversas plataformas de desenvolvimento. Essa biblioteca é responsável por gerir toda a conexão do hardware com o servidor Blynk e gerir as requisições de entrada e saída de dados e comandos. Em nossos projetos, utilizamos as bibliotecas disponíveis na IDE do Arduino, conseguindo realizar o seguinte código:

```
#define BLYNK_PRINT Serial
#define ligar_motor 32
#define desligar_motor 33

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "chave de autenticação enviada pelo aplicativo";
char ssid[] = "Sua net";
char pass[] = "Senha da sua net";

WidgetLCD lcd(V0); //Funcionamento do LCD

BLYNK_CONNECTED() { // Funcionamento dos botões
  Blynk.syncVirtual(V1);
  Blynk.virtualWrite(V1, LOW);
  digitalWrite(desligar_motor, HIGH);
  digitalWrite(ligar_motor, LOW);
  lcd.clear();
  lcd.print(0, 0, "Seja bem-vindo!");
  lcd.print(0, 1, "Motor Control");
}

BLYNK_WRITE(V1) // Ligar o motor
{
  int pinValue = param.asInt();
  if (pinValue == 1) {
    digitalWrite(ligar_motor, HIGH);
    digitalWrite(desligar_motor, LOW);
    lcd.clear();
    lcd.print(2, 0, "Motor Ligado");
  } else {
```

```

        digitalWrite(ligar_motor, LOW);
        digitalWrite(desligar_motor, HIGH);
        lcd.clear();
        lcd.print(0, 0, "Motor Desligado");
    }
};

void setup()
{
    // Debug console
    Serial.begin(9600);
    pinMode(ligar_motor, OUTPUT);
    pinMode(desligar_motor, OUTPUT);

    Blynk.begin(auth, ssid, pass);
}

void loop()
{
    Blynk.run();
}

```

Explicando alguns aspectos desse código, começamos primeiro pelo (*include <Blynk-SimpleEsp32.h>*). Esse include adiciona a biblioteca do Blynk ao projeto que estamos desenvolvendo pela plataforma do Arduino. Logo em seguida, temos o código (*WidgetLCD lcd(V0)*), responsável pela inicialização do LCD no aplicativo, como também, abre a possibilidade de se usar as função de `lcd.print()` e `lcd.clear()`.

Já a função `BLYNK_CONNECTED()` só será acionada quando houver a conexão entre a plataforma criada no aplicativo e a placa de desenvolvimento.

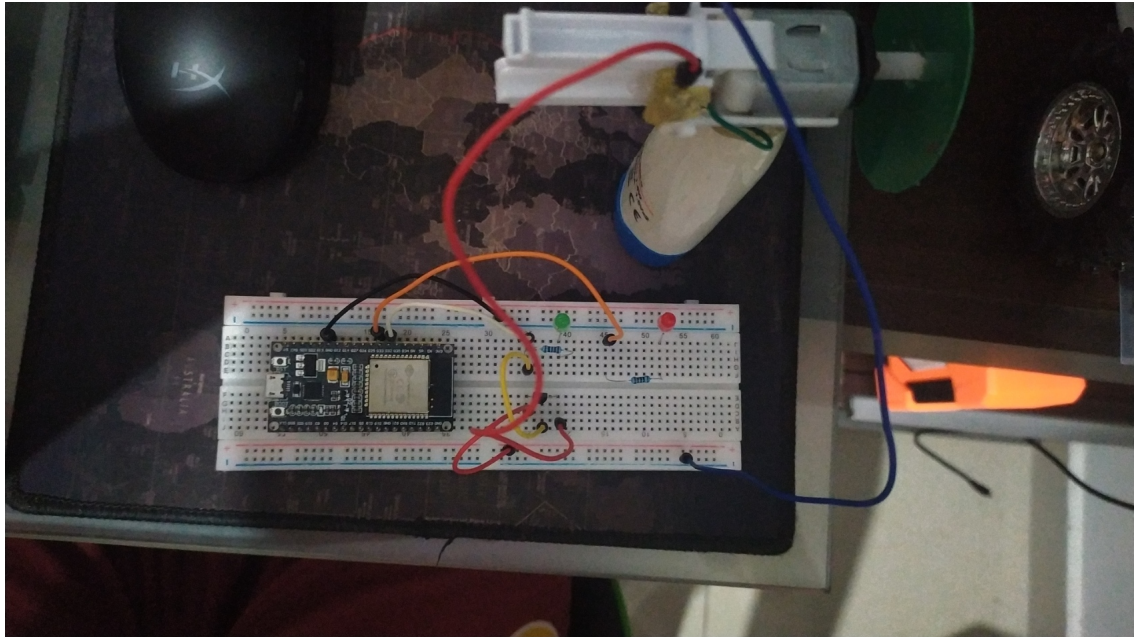
As funções `Blynk.syncVirtual()` e `Blynk.virtualWrite()`, servem ambas para conseguirmos interagir com as portas virtuais do aplicativo. A primeira serve para inicializarmos a porta, e a segunda para conseguirmos interagir com a mesma, mandou valores que desejamos para essa porta virtual, os valores permitidos podem ser: inteiros, strings, booleanos entre outros dependendo widget que aquela porta virtual pertence.

Por ultimo temos a função `Blynk.begin()` que aceita três parâmetros para iniciar a sincronização com o aplicativo, primeiro e o código de autenticação da aplicação realizada, segundo o nome da sua rede que ele deve se conectar e o terceiro a senha da sua rede. A outra função que falta é a `Blynk.run()`, é a partir dela, que toda a aplicação é rodada e iniciada.

### 3 RESULTADOS

Nessa secção apresentarei os resultado conseguidos com o controlador de luminosidade. A figura 6 mostra como ficou o resultado do projeto montado em uma protoboard. Na imagem podemos visualizar o esp-32, os LEDs apresentando o modo de operação e o motor.

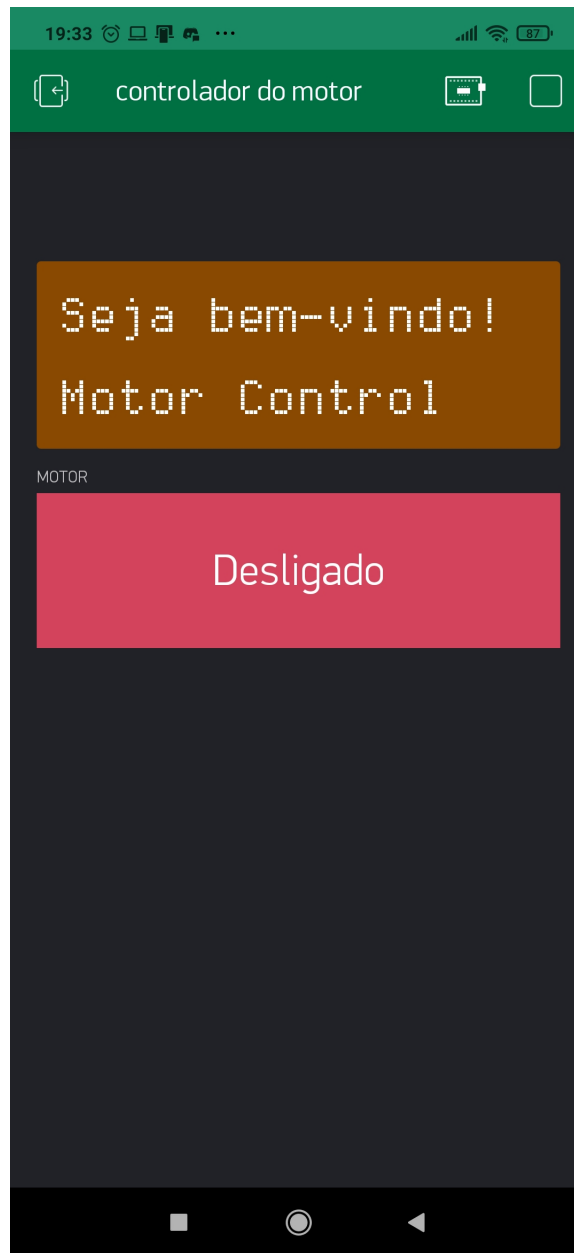
Figura 6: Modelo montado em protoboard



Fonte: Autoria própria

Em seguida, podemos ver o resultado visual na figura 7 do aplicativo assim que começamos a rodar e ele sincroniza toda a aplicação.

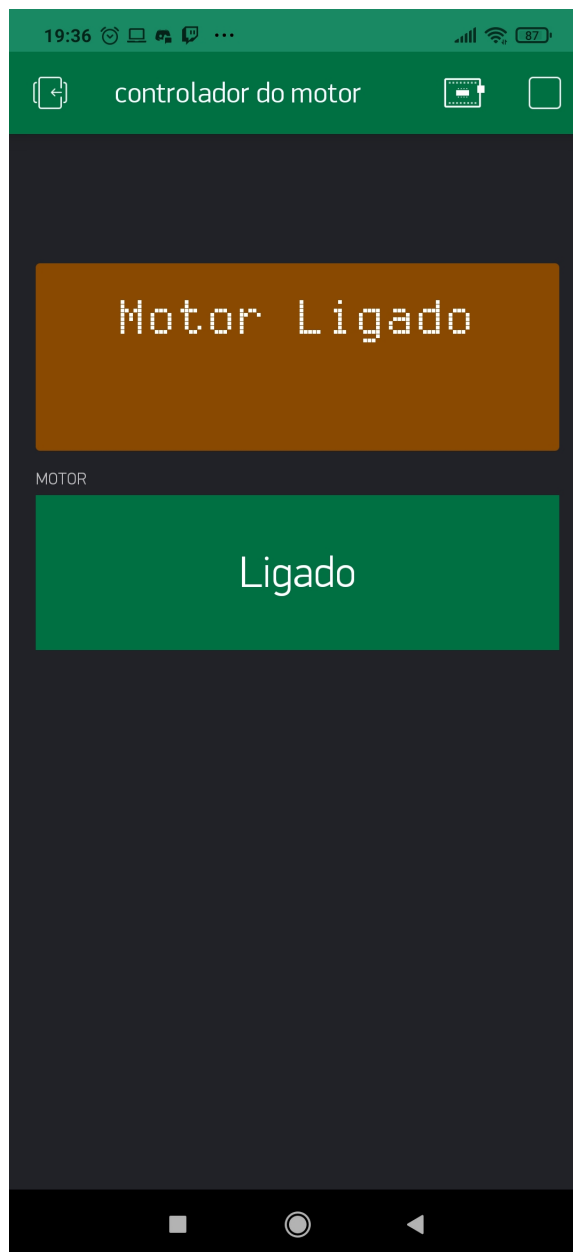
Figura 7: Tela inicial do aplicação



Fonte: Autoria própria

A partir daqui, mostraremos o resultado do acionamento do motor, como também, como ficou na protoboard os alertas visuais. Primeiramente, quando apertamos no botão chamado motor na aplicação, o LCD apresentara a mensagem informando que o motor foi ligado, como também o botão mudar de cor apresentando ligado como nome. Já no circuito, temos o acionamento do motor e o LED verde sendo acesso. Como esperado, conseguimos chegar na solução proposta de maneira visível nas figuras 8 e 9.

Figura 8: Tela modo ligado do aplicação



Fonte: Autoria própria

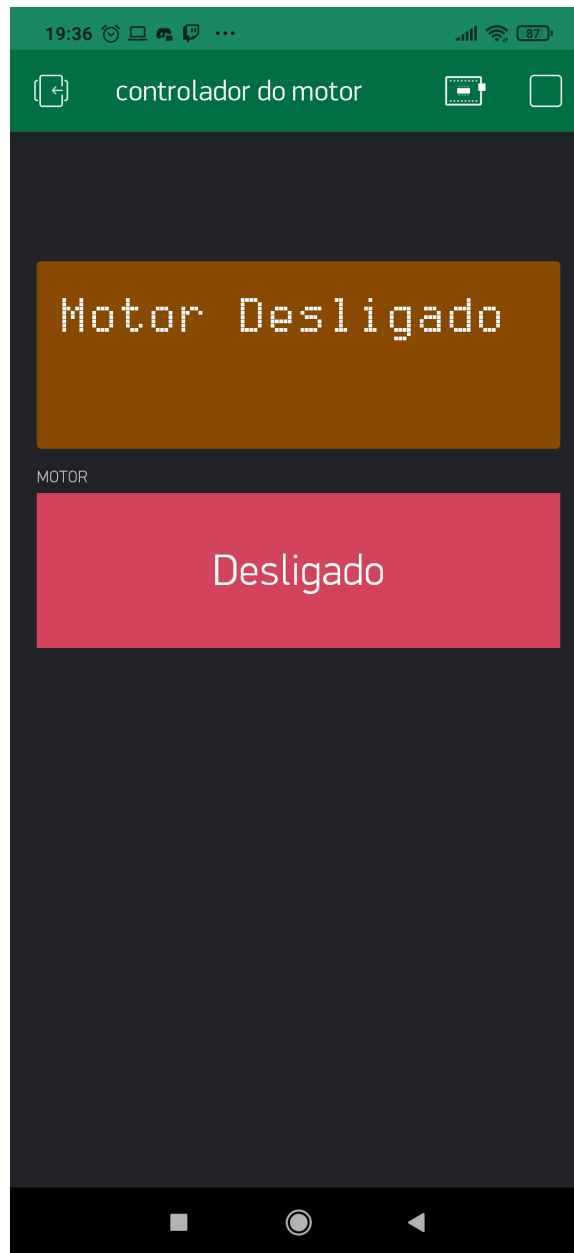
Figura 9: Circuito com motor ligado



Fonte: Autoria própria

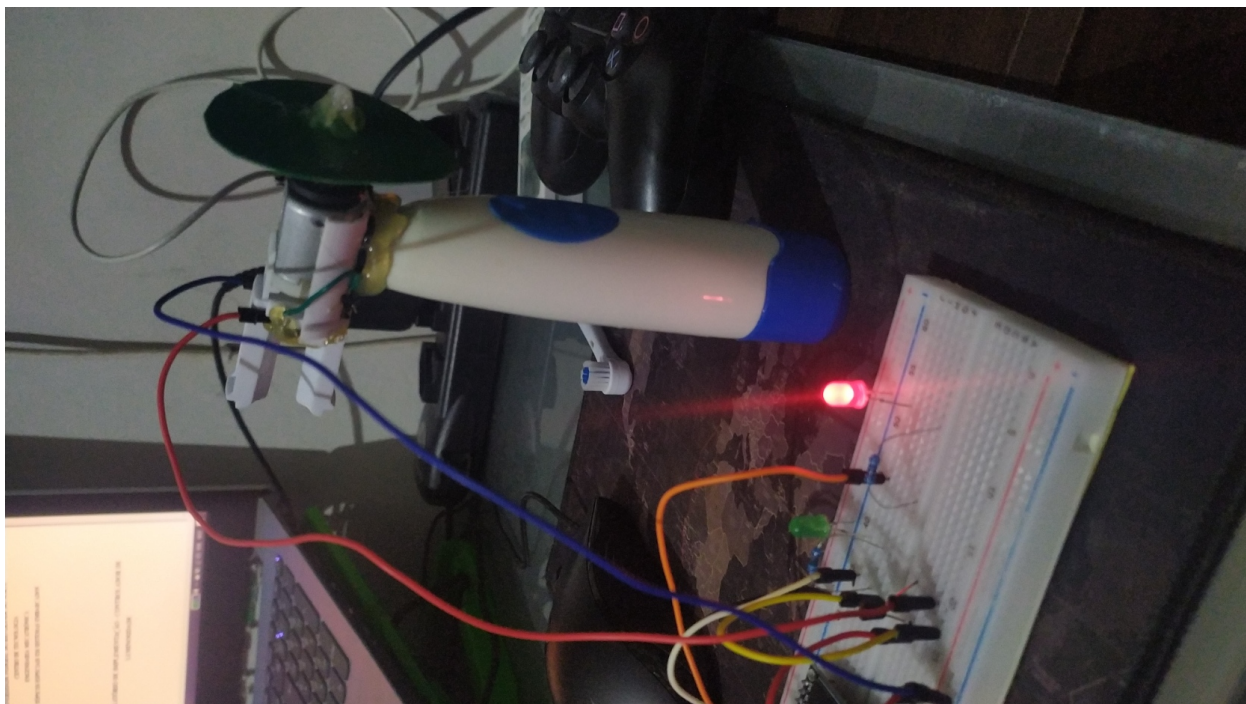
Em contra partida, quando apertávamos o botão novamente, o mesmo deveria voltar a ficar vermelho apresentando como nome Desligado, além de, mostrar no display que o motor está desligado. Já no circuito, o motor ficará desligado e o LED vermelho se acenderá. De mesmo modo, conseguimos chegar na solução proposta de maneira visível nas figuras 10 e 11.

Figura 10: Tela modo desligado da aplicação



Fonte: Autoria própria

Figura 11: Circuito com motor desligado



Fonte: Autoria própria

Portanto, podemos destacar que o controlador funcionou como esperado pelo projeto e projetista. Para uma melhor visualização do funcionamento como um todo, um vídeo de demonstração das funcionalidades desse projeto pode ser acessado através do link: [https://drive.google.com/file/d/1IWf\\_kEq0s4stDRjXIID9OXGvDHXVPrAr/view?usp=sharing](https://drive.google.com/file/d/1IWf_kEq0s4stDRjXIID9OXGvDHXVPrAr/view?usp=sharing)



## **4 CONCLUSÃO**

Dado os ensinamentos em sala de aula e os exemplos encontrados na IDE do Arduíno, conseguir implementar o acionador do motor via Blynk com as funcionalidades esperadas. podendo, não só no futuro gerar outras funcionalidades para esse controlador, como também, poder usar outros periféricos disponibilizados.

## Referências

- 1 DOCUMENTAÇÃO BLYNK. Disponível em: <https://blynk.io/>. Acesso em: 28 abril. 2021.
- 2 DATASHEET ESP-Wroom-32. [S. l.], 2019. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1179101/ESPRESSIF/ESP-WROOM-32.html>. Acesso em: 29 mar. 2021.