

Desenvolvimento de um Carro com Braço Robótico com ESP8266 NodeMCU Controlado por Aplicativo

José Augusto F. Santos¹, Silvio Fernandes. C. Neto², Hermes S. Cavalcante³ e Daniel dos Anjos Costa⁴

¹Instituto Federal da Bahia (IFBA) – Euclides da Cunha – BA – Brasil

²Instituto Federal da Bahia (IFBA) – Euclides da Cunha – BA – Brasil

³Instituto Federal do Sertão Pernambucano (IFSERTÃO-PE) – Petrolina – PE – Brasil

⁴Instituto Federal da Bahia (IFBA) – Santo Amaro – BA – Brasil

jdafrancasantos@gmail.com, silvio.deto@gmail.com,
hermes.siqueira@ifsertao-pe.edu.br e daniel.anjos@ifba.edu.br

Abstract. This work addresses the implementation of a robotic arm attached to a car, remotely controlled via a mobile application or web interface, using the ESP8266 NodeMCU controller. Remote control of the robotic system was possible thanks to the mediation of the Blynk IoT platform and its technologies, with the use of virtual pins that store the state of the control variables. The motivation for this project arises from the need to replace human intervention in dangerous or stressful tasks, such as disarming bombs and exploring risky environments, aiming at greater safety and efficiency.

Resumo. Este trabalho aborda a implementação de um braço robótico acoplado a um carro, controlado remotamente via aplicativo móvel ou interface web, utilizando o controlador ESP8266 NodeMCU. O controle remoto do sistema robótico foi possível graças à mediação da plataforma Blynk IoT e as suas tecnologias, com o uso de pinos virtuais que armazenam o estado das variáveis de controle. A motivação para este projeto surge da necessidade de substituir a intervenção humana em tarefas perigosas ou desgastantes, como desarme de bombas e exploração de ambientes de risco, visando maior segurança e eficiência.

1. Introdução

A evolução da computação representa um marco crucial no desenvolvimento tecnológico da civilização moderna. Ela não só aumentou a capacidade e a velocidade dos sistemas, mas também possibilitou o surgimento de tecnologias emergentes, como robótica e Internet das Coisas (IoT). A robótica é a área que se dedica ao desenvolvimento de sistemas programáveis capazes de manipular objetos do ambiente físico e executar tarefas complexas de maneira autônoma [Matarić 2014].

A Internet das Coisas (IoT) é uma parte integrada da futura Internet e pode ser definida como uma infraestrutura de rede global dinâmica com capacidades de auto configuração baseadas em protocolos de comunicação padrão e interoperáveis, onde as “coisas” físicas e virtuais têm identidades, atributos físicos e personalidades virtuais e usam interfaces inteligentes, e são perfeitamente integradas à rede de informações [Yehia et al., 2015].

A fusão entre robótica e Internet das Coisas (IoT) está transformando a maneira como sistemas autônomos funcionam e interagem com o ambiente. A união da habilidade dos robôs em executar tarefas complexas e físicas com a conectividade

proporcionada pela IoT resulta em um ecossistema dinâmico e inteligente. Essa integração permite que os robôs coletem e transmitam dados em tempo real, ajustem-se a mudanças no ambiente e colaborem de forma eficaz com outros dispositivos conectados [Ray, 2016].

Neste trabalho desenvolvemos um carro com braço robótico com o controlador ESP8266 NodeMCU. Esse controlador possibilita o controle remoto do sistema robótico via Wi-Fi, permitindo a transmissão em tempo real dos estados do sistema e o gerenciamento de seus atuadores. O sistema robótico é operado por duas interfaces que funcionam em paralelo: um aplicativo de celular e uma interface web.

2. Fundamentação Teórica

Esta seção descreve as tecnologias de hardware e software utilizadas no projeto. Para o desenvolvimento dos mecanismos de controle e movimentação mecânica do carro e do braço robótico, foram utilizados os materiais e tecnologias mencionados na Tabela 1.

Tabela 1. Itens e Tecnologias usadas para construção do projeto

Item	Definição
ESP8266 NodeMCU	O NodeMCU (<i>Node Micro Controller Unit</i>) é uma plataforma de prototipação com um firmware de código aberto construído em torno de um System-on-a-Chip (SoC) chamado ESP8266. Possui 17 pinos GPIO (<i>General-Purpose Input-Output</i>), que são pinos de entrada e saída de dados - ou sinais elétricos - para propósitos gerais, além de elementos de processamento, armazenamento, entrada, saída e Wi-Fi [Make-IT 2022].
Blynk IoT	A plataforma de prototipação Blink IoT oferece serviços personalizáveis para controle remoto de hardware programável, permitindo uma gestão eficiente e flexível dos dispositivos. Além disso, a plataforma facilita a coleta em tempo real de indicadores de estado do sistema, integrando-se de forma prática a projetos de hardware e software embarcados [Serrano 2018].
Ponte H L298N	A ponte H atua como um amplificador de potência, permitindo que o sinal de controle de baixa potência seja usado para controlar a corrente de alta potência que flui através do motor CC. Existem diversos circuitos de Ponte H disponíveis no mercado; no nosso projeto, utilizamos o CI L298N, amplamente empregado em conjunto com controladores como Arduino e ESP.
Servomotor	É um tipo especial de motor que é usado para movimentar peças possibilitando o controle da velocidade de giro e do ângulo de rotação do seu eixo. Um servomotor é um dispositivo eletromecânico que combina um motor elétrico com um sistema de controle de posição [Boylestad 2013]. Segundo Mattede (2014), para controlar a velocidade e a posição final do eixo do motor é usado um mecanismo de <i>feedback</i> (realimentação) de posição. Esse mecanismo de <i>feedback</i> de posição informa em tempo real qual é a posição atual do eixo do servomotor.
Motor de Corrente Contínua	Convertem energia elétrica em energia mecânica, proporcionando movimento rotativo. Uma de suas principais características é a capacidade de controlar a velocidade e a direção do motor variando a magnitude e a direção da corrente elétrica que passa pelos enrolamentos do rotor [Boylestad 2013].

2.1 Trabalhos relacionados

Diversos projetos semelhantes ao desenvolvido foram identificados na literatura, porém, nenhum integrou todas as tecnologias mencionadas neste trabalho. Destacaremos apenas três desses projetos para análise.

Souza et. al. (2023) desenvolveu um braço robótico com controle multimodal usando o controlador ESP8266 NodeMCU. O sistema desenvolvido contou com 3 mecanismos de controle paralelos e hierárquicos, a saber: controle local com botões físicos na placa, controle remoto via interface web ou aplicativo e um controle de voz realizado com o auxílio do Google Assistente. A plataforma Blynk IoT foi usada tanto para mandar os comandos de voz do Google assistente e transferir os dados para o controlador, quanto para realizar o mesmo processo com a interface web ou aplicativo, e assim movimentar o braço. Para os botões, um circuito integrado codificador foi utilizado para reduzir o número de pinos necessários para o funcionamento.

No segundo trabalho foi desenvolvido um braço robótico controlado por arduino e um joystick na interação entre homem e máquina [Santos et. al. 2023]. Os autores deixam claro que a motivação do trabalho se baseia no fato do Joystick possuir uma interface de comandos bem consolidada e que pode ser usada para uma grande variedade de aplicações.

No trabalho de Barbosa e Silva (2024) foi desenvolvido um braço robótico controlado via celular com comandos seriais enviados por interface Bluetooth. O braço robótico é controlado por seis servomotores, cuja movimentação é ajustada proporcionalmente por meio de um aplicativo desenvolvido para dispositivos Android. Este aplicativo envia, via Bluetooth, pacotes de valores variando de 0 a 180 que correspondem ao ângulo de rotação de cada servomotor. Esses valores são então interpretados pelo código no Arduino, que gera os sinais necessários para movimentar o braço robótico de acordo com os comandos recebidos.

Nosso trabalho apresenta várias semelhanças com os projetos mencionados anteriormente, especialmente no desenvolvimento do braço mecânico. Entretanto, o diferencial deste projeto reside em seu sistema compacto e autônomo, que dispensa o uso de uma fonte externa, permitindo a movimentação livre do protótipo com o carro. Além disso, foram integradas tecnologias adicionais, como motores de corrente contínua (CC) e uma Ponte H, que viabilizam a movimentação eficiente do veículo.

3. Desenvolvimento

Nesta seção apresentaremos os detalhes de implementação do projeto, explicando as tecnologias e ferramentas utilizadas no sistema.

O sistema como um todo é formado por um Aplicativo de celular e uma Interface Web que atuam em paralelo e enviam comandos de atuação para um Hardware Controlador ESP8266 via internet com mediação da Plataforma Blynk IoT. O Hardware Controlador ao receber diretrizes envia sinais elétricos para acionar o movimento dos Servomotores que atuam no Braço Robótico ou sinais elétricos para acionar os Motores de corrente contínua do carro robótico. A ilustração da montagem do sistema está apresentada na Figura 01 abaixo.

O ESP8266 NodeMCU possui uma memória flash que permite embarcar códigos de controle, além de pinos atuadores que podem ser de entrada/saída digital ou analógica. Além disso, o ESP8266 possui um módulo de conexão Wi-Fi integrado, que lhe confere a capacidade de se comunicar com outros dispositivos por meio de uma rede sem fio. Apesar do dispositivo ESP8266 NodeMCU possuir mais de 23 portas, nem todas podem ser usadas como entrada e saída de dados comuns [Make-IT 2022]. Algumas têm funções específicas para atuação em alguns dispositivos eletrônicos, como: comunicação e carregamento de dados em memória flash externa, comunicação utilizando protocolos fixos, etc. Por essa limitação, utilizamos no projeto 4 portas digitais para saída dos sinais de controle do carro robótico e outras 4 portas digitais para controle dos servomotores de atuação no braço robótico.

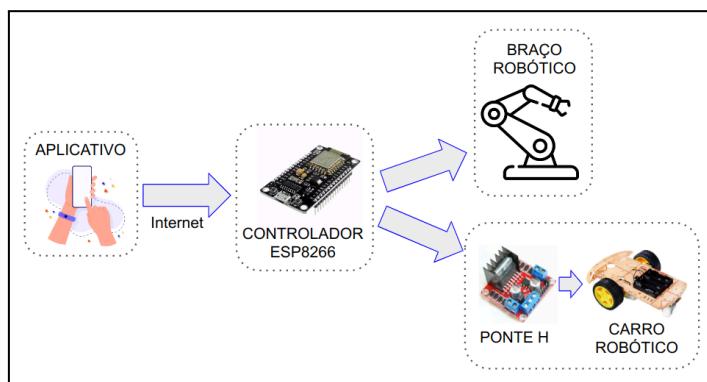


Figura 1. Organização esquemática dos controles

Para a movimentação do braço utilizamos servomotores, que são uma categoria de motores que permite o controle do ângulo de rotação de seu eixo. Os servomotores usados no projeto necessitam de uma alimentação de 5 volts nos seus terminais de alimentação, bem como sinais de controle em seu terceiro terminal. Os sinais elétricos recebidos no terceiro pino dos servomotores são de baixa potência e são enviados diretamente pelo controlador ESP8266. Esses sinais indicam o novo ângulo de rotação para o eixo do servomotor. Cada servomotor movimenta mecanicamente uma das quatro partes móveis do braço.

Por fim, para a movimentação do carro robótico utilizamos dois motores de corrente contínua (CC). Esses motores trabalham de forma diferente dos servomotores pois não tem um ângulo limite de atuação, ou seja, seu eixo pode girar sem limitação mecânica.

O controlador ESP8266 não consegue fornecer potência suficiente para controlar diretamente os dois motores CC e por causa disso, adicionamos ao projeto um circuito amplificador de potência conhecido como ponte H. A ponte H vai receber o sinal de baixa potência do Controlador e vai amplificar na medida para controlar a direção e a velocidade do carro robótico.

A montagem mecânica das peças do braço robótico utilizado no projeto é de domínio público e foi retirada do blog Ultimaker Thingiverse [Gray 2014], conforme apresentado na Figura 02a. As peças do braço foram confeccionadas em MDF de 3mm

com corte a laser. Para a montagem do carro robótico, pesquisamos modelos de chassis e peças que poderíamos confeccionar, assim como fizemos no desenvolvimento do braço robótico. No entanto, optamos por adquirir um kit com todas as peças necessárias. A ilustração da montagem do carro e do braço é exibido na Figura 02b

3.1. Montagem dos circuitos eletrônicos do sistema

Na Figura 02c apresentamos o esquema de montagem dos circuitos eletrônicos do sistema. No centro da figura temos o ESP8266 que comanda todas ações do sistema. Ele recebe comandos da interface gráfica (aplicativo de celular ou interface web) por intermédio dos pinos virtuais explicados nas seções adiante. Os valores dos pinos indicam a ação que deve ser realizada no sistema.

Na parte inferior da Figura 02c podemos observar a presença de quatro servomotores nomeados com: M0, M1, M2 e M3. Cada um desses servomotores atua em uma região do braço robótico. O servomotor M0 na região nomeada como “Base”, o M1 na região nomeada como “Quadrado”, o M2 na região nomeada como “Garra” e o M3 na região nomeada como “Trapézio”, conforme pode ser conferido na Figura 02a.

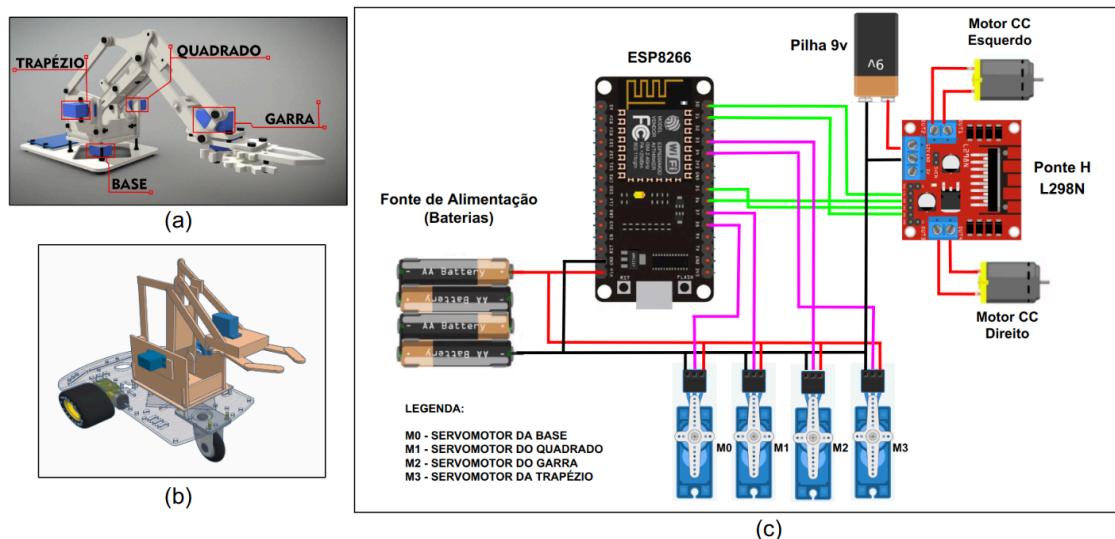


Figura 02 - Esquema de montagem do circuito eletrônico do Sistema.

Os quatro servomotores recebem sinal de controle do ESP8266 obedecendo a seguinte relação: Servo M0 ligado ao pino D8, Servo M1 ligado ao pino D7, Servo M2 ligado ao pino D3 e Servo M3 ligado ao pino D4. Além do sinal de controle, os servomotores também recebem alimentação vinda das baterias acopladas ao sistema. Estas baterias permitem a total autonomia do sistema de outras fontes externas.

Na região direita da Figura 02c podemos observar o circuito conhecido como Ponte H e os dois motores CC, um para a roda da direita e outro para a roda da esquerda do carro. É importante destacar que o ESP8266 foi programado para controlar os motores CC com Modulação por Largura de Pulso (PWM - *Pulse Width Modulation*). Essa técnica permite controlar a velocidade de giro dos motores sem perder o torque de

força. Entretanto, apesar do ESP8266 emitir o sinal de controle correto nas suas portas de saída, os mesmos não têm potência suficiente para colocar em operação os motores CC e por isso usou-se o componente chamado Ponte H. A alimentação da ponte H e dos motores CC foi realizada com uma bateria de 9V por causa da demanda de potência desses componentes. É importante destacar que foi utilizado duas fontes de alimentação no sistema e por isso unificamos os pólos negativos para evitar o efeito de flutuação de tensão.

Na montagem física, utilizamos as portas de saída D0 e D1 do ESP8266 conectadas às portas de entrada IN1 e IN2 da ponte H, respectivamente, para controlar o motor da roda esquerda. As portas de saída D5 e D6 do ESP8266 foram conectadas às portas de entrada IN3 e IN4 da ponte H, respectivamente, para controlar o motor da roda direita. O motor da esquerda foi ligado nas portas de saída OUT1 e OUT2 da ponte H e o motor da direita nas portas de saída OUT3 e OUT4.

3.2. Montagem lógica do sistema

Nesta seção, daremos continuidade na explicação dos circuitos elétricos e de controle do sistema. Também informamos que o sistema pode ser controlado por um Aplicativo de Celular ou uma Interface Gráfica Web que atuam em paralelo. As duas interfaces de controle possuem botões comuns de acionamento e Sliders (ou "controle deslizante") que são componentes interativos que permitem ao usuário selecionar um valor ajustável em um intervalo contínuo. As Figuras 03a e 03b exibem as interfaces gráficas dos softwares.

Podemos observar, na parte superior das figuras, o uso de quatro Sliders vinculados aos quatro servomotores que movimentam o braço robótico. Eles permitem que o usuário selecione um valor dentro de um intervalo limitado de ângulo de rotação para os servos. Na parte Inferior das figuras foram utilizados cinco botões para controlar a movimentação do carro robótico com as seguintes identificações: "Frente", "Esquerda", "Direita", "Ré" e "Parar".

Os botões têm acionamento simples com um clique, porém, com funcionamento conjugado como um botão do tipo rádio. Se o usuário pressionar o botão "Frente" o veículo receberá um comando para se movimentar para frente, e se em seguida o usuário pressionar o botão "Ré", o botão "Frente" será desativado automaticamente e o veículo mudará a direção de movimentação. O uso desses botões garantem que não haverá sobreposição de comandos.

Cada componente da interface gráfica está vinculado a um Pino Virtual na plataforma Blynk IoT que armazenará valores passados pela interface e repassará os mesmos para o controlador ESP8266. Os pinos foram nomeados com V0, V1, V2 e V3 foram vinculados aos sliders "Base", "Quadrado", "Garra" e "Trapézio" respectivamente.

Por outro lado, foi utilizado apenas um pino virtual para controlar a movimentação dos motores CC do carro robótico, nomeado com "V4". Acionando qualquer um dos cinco botões da interface gráfica de controle do carro um valor inteiro discreto é repassado para o pino virtual V4. Esse valor indicará ao ESP8266 a ação que deverá ser executada pelos motores CC.



Figura 03 - (a) Interface do aplicativo de celular; (b) Interface web.

Na Figura 04 apresentamos a primeira parte do código de controle. Nas linhas 2, 3 e 4 exibimos os comandos para estabelecer a conexão entre o dispositivo e o projeto na plataforma Blynk IoT. Na linha 2 o comando define o ID do template do projeto no Blynk, na linha 3 define o nome do template do projeto e o comando da linha 4 define o token de autenticação do projeto Blynk. O token de autenticação é uma chave única gerada pela plataforma Blynk para cada projeto. Ele é utilizado para autenticar e autorizar o dispositivo ESP8266 a se conectar ao servidor Blynk e interagir com o projeto específico. Sem este token, o dispositivo não poderá se conectar ao servidor Blynk. É importante destacar que os valores apresentados na Figura 04 não são reais do nosso projeto.

```

1  #define BLYNK_TEMPLATE_ID "ID DO PROJETO"           // Não são valores reais
2  #define BLYNK_TEMPLATE_NAME "SISTEMA ROBÓTICO"
3  #define BLYNK_AUTH_TOKEN "TOKEN DO BLYNK"
4
5
6  #define BLYNK_PRINT Serial
7  #include <ESP8266WiFi.h>
8  #include <BlynkSimpleEsp8266.h>
9  #include <Servo.h>
10
11 char auth[] = BLYNK_AUTH_TOKEN;
12 char ssid[] = "SSID_REDE"; // Não são valores reais
13 char pass[] = "SENHA_REDE";
14
15 Servo M0, M1, M2, M3 // Criação dos objetos Servomotor
16
17 // Variáveis com o ângulo de movimentação dos Servos
18 int anguloM0, anguloM1, anguloM2, anguloM3 ;
19

```

Figura 04 - Parte I do código de controle.

Nas linhas 12 e 13 apresentamos as variáveis que armazenam as informações de SSID e SENHA da rede Wi-Fi que o dispositivo ESP8266 vai se conectar. Na linha 15 exibimos o instanciamento dos objetos “Servo” que utilizaremos para controlar os servomotores do braço robótico. E por fim na linha 18 criamos 4 variáveis para armazenar o ângulo dos servomotores.

Na Figura 05a apresentamos a segunda parte do código de controle do sistema. Seu código controla a movimentação dos servomotores com base nas notificações recebidas pelos pinos virtuais V0, V1, V2 e V3. Por exemplo, entre as linhas 62 e 66

tem a função “BLYNK_WRITE(V0)” que recebe as notificações de alteração de estado do pino V0 provocado pelo usuário no aplicativo de celular ou na interface web. Ao receber uma notificação com o novo valor de ângulo para o servo M0 essa função controlará a movimentação do eixo de M0 até alcançar o ângulo desejado. O mesmo procedimento se repete nas demais funções “BLYNK_WRITE” presentes na Figura 05a.

Por fim, na Figura 05b apresentamos a última parte do código de controle do sistema dentro da função “BLYNK_WRITE(V4)”. Essa função recebe notificações do pino V4 que reservamos para o controle do carro robótico. Na linha 89 usamos o comando “param.toInt()” para salvar na variável “direcao” o valor transferido pelo pino V4. Nas linhas seguintes atribuímos uma sequência de condicionais que controlam a movimentação do veículo.

No primeiro condicional (linha 91) se o valor de “direcao” for igual a 1 os motores CC do carro serão acionados para movimentá-lo para frente com velocidade constante. Os dois motores receberão a mesma velocidade de atuação, o que implica em uma movimentação retilínea. No segundo condicional (linha 96) se o valor de “direcao” for igual a 2 os motores serão acionados no sentido inverso do primeiro condicional, resultando na movimentação para trás do carro. É importante destacar que para manter a movimentação reta os dois motores receberão a mesma velocidade.

```

87  BLYNK_WRITE(V4)
88  {
89      int direcao = param.toInt();
90
91      if(direcao == 1)
92      {
93          // Indica que o carro deve movimentar para frente
94          // Os dois motores funcionam na mesma velocidade
95      }
96      else if(direcao == 2)
97      {
98          // Indica que o carro deve movimentar para trás
99          // Os dois motores funcionam na mesma velocidade, mas na rotação inversa
100     }
101    else if(direcao == 3)
102    {
103        // Indica que o carro deve movimentar para frente e esquerda
104        // O motor da esquerda vai girar com uma velocidade reduzida
105    }
106    else if(direcao == 4)
107    {
108        // Indica que o carro deve movimentar para frente e direita
109        // O motor da direita vai girar com uma velocidade reduzida
110    }
111    else
112    {
113        digitalWrite(D0, LOW); // Motor CC parado
114        digitalWrite(D1, LOW); // Motor CC parado
115        digitalWrite(D5, LOW); // Motor CC parado
116        digitalWrite(D6, LOW); // Motor CC parado
117    }
118 }
119
120
121 void loop()
122 {
123     Blynk.run();
124 }
125

```

(a)

(b)

Figura 05 - (a) Parte II do código e (b) Parte III do código de controle.

No terceiro condicional (linha 101) se “direcao” for igual a 3 o carro deve se movimentar para frente e para esquerda. Para isso, diferenciamos as velocidades dos motores. Ao fazer a curva para a esquerda, a velocidade do motor do lado esquerdo é reduzida em comparação à do lado direito, resultando em uma rotação menor do motor esquerdo. Essa diferença de rotação faz com que o veículo se desloque para frente e para a esquerda.

O mesmo procedimento é realizado no quarto condicional (linha 106) porém, com a lógica invertida. O quarto condicional sinaliza que o veículo deve se movimentar para frente e direita, logo o motor do lado direito deve girar menos que o motor do lado esquerdo.

Por fim, o “else” (linha 111) possui quatro comandos com a função “digitalWrite” que estabelecem valor lógico LOW para os pinos de saída usados para comandar os motores CC. Esses quatro comandos param a movimentação do veículo.

4. Limitações do trabalho

O projeto tem um enorme potencial e conseguimos desenvolver tudo que foi previamente estabelecido. Entretanto, por se tratar de um projeto com limitações orçamentárias e de material, tivemos que adaptar diversas peças, o que não impede o funcionamento do sistema. Adicionamos à lógica de programação algumas limitações na movimentação do braço robótico para evitar choques com outras partes do sistema.

Além disso, o braço robótico é capaz de realizar tarefas como agarrar e soltar objetos, enquanto o carro pode se mover para frente, para trás e fazer curvas para a esquerda e para a direita. No entanto, o sistema não é capaz de realizar movimentos extremamente precisos ou complexos devido às limitações dos componentes utilizados.

Uma segunda limitação do projeto é a dependência da rede Wi-Fi. O controle do sistema através das interfaces gráficas requer uma conexão estável com a internet. Se a rede Wi-Fi estiver instável ou indisponível, o controle remoto do sistema pode ser comprometido, limitando as operações do braço robótico e do carro. Apesar dessas limitações, o projeto demonstrou ser funcional e oferece uma base sólida para futuros desenvolvimentos e melhorias. A integração das interfaces gráficas com o controlador ESP8266 através da plataforma Blynk IoT mostrou-se eficaz e abre caminho para a expansão das capacidades do sistema.

Para ilustrar o funcionamento do sistema robótico desenvolvido, criamos um vídeo exemplificativo disponível no seguinte link do Youtube: <https://www.youtube.com/watch?v=TxxQuabMiik>.

5. Conclusão

Neste trabalho desenvolvemos um carro com braço robótico com o controlador ESP8266 NodeMCU. Esse controlador possibilita o controle remoto do sistema robótico via Wi-Fi, permitindo a transmissão em tempo real dos estados do sistema e o gerenciamento de seus atuadores. O sistema robótico é operado por duas interfaces que funcionam em paralelo: um aplicativo de celular e uma interface web. Nas seções anteriores, apresentamos todos os detalhes da montagem mecânica, eletrônica e lógica do sistema.

A motivação para este projeto decorre da necessidade de substituir a intervenção humana em tarefas altamente perigosas e fisicamente desgastantes, como o desarme de bombas e a exploração de ambientes de risco. Ao substituir o trabalho humano por tecnologias avançadas, buscamos aumentar a segurança, proteger vidas e melhorar a eficiência operacional.

Para contribuir com trabalhos futuros, sugerimos as seguintes melhorias: 1) Adicionar sensores de distância ao protótipo para aumentar a segurança durante a execução das tarefas, prevenindo colisões indesejadas; 2) Implementar um controle mais preciso da velocidade, permitindo que o usuário ajuste a velocidade do carro através da interface gráfica; 3) Integrar novos mecanismos de controle, tanto local quanto remoto, utilizando redes de computadores ou a internet, para expandir as possibilidades de operação e facilitar o acesso e monitoramento do sistema robótico. Essas melhorias proporcionarão avanços significativos no projeto, oferecendo maior segurança, funcionalidade e controle.

6. Referências

BARBOSA, Wanderson H. A.; SILVA, Rafael, de O. Braço articulado comandado via bluetooth por um aplicativo desenvolvido na plataforma android. 17 f. Graduando em engenharia Mecânica - Universidade de Rio Verde, UNIVR, GO. Disponível em: <https://firewall03.unirv.edu.br/conteudos/fckfiles/files/Wanderson%20Henrique%20Alves%20Barbosa.pdf>. Acesso em: 3 ago. 2024.

BOYLESTAD, R. **Introdução à Análise de Circuitos**. 12^a ed. LTC, 2013.

GRAY, Ben. MeArm V0.4 - Pocket Sized Robot Arm. In: Ultimaker thingverse. [S. l.], [2014]. Disponível em: <https://www.thingiverse.com/thing:360108/files>. Acesso em: 26 set. 2024.

MAKE-IT. **Details and Specifications, Overview and Setting Up Your NodeMCU**. Disponível em: <https://www.make-it.ca/nodemcu-details-specifications/>. Acesso em: 26 set. 2024.

MATARIĆ, M. J. (2014). Introdução à robótica. Editora Blucher.

MATTEDE, H. **Motor de corrente contínua, características e aplicações! Mundo da eletrônica**, 2014b. Disponível em: <https://www.mundodaeletrica.com.br/motor-de-corrente-continua-caracteristicas-e-aplicacoes/>. Acesso em: 26 set. 2024.

YEHIA, Lobna et al. Hybrid security techniques for Internet of Things healthcare applications. *Advances in Internet of Things*, v. 5, n. 03, p. 21, 2015.

RAY, Partha Pratim. Internet of robotic things: concept, technologies, and challenges. *IEEE access*, v. 4, p. 9489-9500, 2016.

SANTOS, Dhomini; SILVA, Helena P. de S.; COSTA, Daniel dos A. Desenvolvimento de um Braço Robótico controlado por Joystick. In: *Anais da XXIII Escola Regional de Computação Bahia, Alagoas e Sergipe*. SBC, 2023. p. 82-91.

SERRANO, T. M. **Introdução ao Blynk App 2018**. Disponível em: <https://embarcados.com.br/introducao-ao-blynk-app>. Acesso em: 26 set. 2024.

SOUZA, Joedson; FERREIRA, Magnum S.; COSTA, Daniel A.. Desenvolvimento de um Braço Robótico com Controle Multimodal Usando o Controlador ESP8266 NodeMCU. In: *Anais da XXIII Escola Regional de Computação Bahia, Alagoas e Sergipe*. SBC, 2023. p. 72-81.