

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENG. DE CONTROLE E AUTOMAÇÃO

LEONARDO ZAMBOTTO DA SILVA - 00262538

**PROJETO DE PLATAFORMA
EDUCACIONAL DE
DESENVOLVIMENTO ROBÓTICA**

Porto Alegre
2021

LEONARDO ZAMBOTTO DA SILVA - 00262538

**PROJETO DE PLATAFORMA
EDUCACIONAL DE
DESENVOLVIMENTO ROBÓTICA**

Trabalho de Conclusão de Curso (TCC-CCA) apresentado à COMGRAD-CCA da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de *Bacharel em Eng. de Controle e Automação*.

ORIENTADOR:

Prof. Dr. Rafael Antônio Comparsi Laranja

Porto Alegre
2021

LEONARDO ZAMBOTTO DA SILVA - 00262538

**PROJETO DE PLATAFORMA
EDUCACIONAL DE
DESENVOLVIMENTO ROBÓTICA**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Disciplina de TCC do curso *Eng. de Controle e Automação* e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Rafael Antônio Comparsi Laranja, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil)

Banca Examinadora:

Prof. Dr. Rafael Antônio Comparsi Laranja, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil)

Prof. Dr. Mário Roland Sobczyk Sobrinho, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil)

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS
Doutor pela Universidade Federal de Minas Gerais, Belo Horizonte, Brasil)

Mário Roland Sobczyk Sobrinho
Coordenador de Curso
Eng. de Controle e Automação

Porto Alegre, novembro de 2021.

AGRADECIMENTOS

Agradeço à minha família por todas as oportunidades proporcionadas e pela sempre presente confiança nas minhas capacidades, até nos momentos que essa me faltava.

Em especial agradeço à minha mãe, Elisa Cristina Zambotto, que sempre fez e continua fazendo o dobro pra garantir que eu e meu irmão estejamos bem. Obrigado pelo amor, dedicação, paciência e por estar comigo em todos os momentos, ainda que de longe.

À Universidade Federal do Rio Grande do Sul, UFRGS, por prover-me a capacidade de adquirir conhecimento, de conhecer pessoas maravilhosas e de ampliar meus horizontes e minha visão de mundo nestes anos de graduação.

Ao meu orientador, Professor Rafael A.C. Laranja, por auxiliar nos momentos de dúvida no desenvolvimento deste trabalho e ensinar a sempre levar todos os fatores em consideração, porque tudo "depende".

Aos meus amigos e colegas de curso, por sempre dar aquela força ou motivação que faltava, pelos inesquecíveis momentos vividos e por todas as jogatinas e as sessões de estudo nas madrugadas pré-prova.

RESUMO

Neste trabalho foi apresentado o desenvolvimento de uma plataforma robótica educacional de comando remoto, visando aumentar a motivação de alunos de cursos superiores de graduação. Foram apresentadas as etapas de projeto informacional para a concepção desse sistema, consistindo em uma análise do problema e pesquisa de mercado sobre soluções já existentes. Desenvolveu-se a etapa conceitual do projeto, na qual foram definidos os fatores de projeto. As fases de projeto preliminar e detalhado foram desenvolvidas, nas quais foram realizados processos de seleção dos componentes e protocolo de comunicação utilizados, desenho da estrutura física, cálculo da cinemática inversa, montagem da plataforma desenvolvida em uma configuração selecionada e uma análise da capacidade de reconfiguração do sistema. Ao fim deste trabalho, obteve-se uma plataforma reconfigurável, modular e acessível para discentes em níveis intermediários do curso de graduação, capaz de proporcionar aos mesmos uma atividade prática multidisciplinar.

Palavras-chave: Engenharia, controle e automação, reconfigurável, educacional, robô, quadrúpede, MQTT.

ABSTRACT

This project presents the development of a remote-controlled educational robotic platform, aiming to increase undergraduates' interest and motivation. The informational project stage for this system's conception is shown, including a problem's analysis and a market survey of available solutions. The conceptual design is also developed, by defining the system requirements. The preliminary and detailed project phases were developed, showing the components and communication protocol's selection process, structural design, inverse kinematics calculation, platform assembly and system's interchangeability analysis. By the end of this paper, it is expected to have a reconfigurable, modular and accessible platform for intermediate-level graduate students, providing a hands-on multidisciplinary activity.

Keywords: engineering, robot, reconfigurable, educational, control and automation, quadruped, MQTT.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	7
LISTA DE ABREVIATURAS	8
1 INTRODUÇÃO	9
2 REVISÃO DA LITERATURA	10
2.1 Robótica Educacional	10
2.2 Robôs com Locomoção por Pernas	11
3 MATERIAIS E MÉTODOS	13
3.1 Projeto Informacional	13
3.1.1 Análise do Problema	13
3.1.2 Pesquisa de Mercado	14
3.2 Projeto Conceitual	14
3.2.1 Fatores de Projeto	14
3.3 Projeto Preliminar	15
3.3.1 Seleção do Material	15
3.3.2 Seleção dos Atuadores	15
3.3.3 Seleção do Microcontrolador	17
3.3.4 Seleção de Elementos Elétricos de Potência	17
3.3.5 Seleção do Protocolo de Comunicação	18
3.4 Projeto Detalhado	20
3.4.1 Desenho de Padrão de Encaixe Modular	20
3.4.2 Estudo de Caso	22
3.4.3 Cálculo da Cinemática Inversa	23
4 RESULTADOS	25
4.1 Operação Nominal	25
4.2 Validação de Reconfiguração	25
4.3 Aceitação Discente	29
5 CONCLUSÕES	32
REFERÊNCIAS	33
APÊNDICE A - FORMULÁRIO UTILIZADO NA PESQUISA COM DIS- CENTES	37
APÊNDICE B - CÓDIGO IMPLEMENTADO EM C++	41

LISTA DE ILUSTRAÇÕES

1	Robôs montados com o <i>kit</i> Lego Mindstorms	10
2	Plataformas do projeto EduBOT utilizando metarreciclagem	11
3	Respostas de discentes quanto a motivação com o curso e com atividades práticas	13
4	Configuração utilizada na análise de torque	16
5	Interface do aplicativo utilizado para comunicação via protocolo MQTT	19
6	Padrão de encaixe modular desenvolvido	20
7	Diferentes possibilidades de posicionamento de um atuador	20
8	Peças desenhadas para a plataforma	21
9	Desenho da peça de estrutura principal visando a redução de massa . .	22
10	Configuração de juntas e ligamentos escolhida para estudo de caso . .	22
11	Pose considerada para o cálculo da cinemática inversa	23
12	Função desenvolvida em C++ para o cálculo de cinemática inversa . .	24
13	Montagem da plataforma para estudo de caso	25
14	Configuração de estudo de caso com grau de liberdade adicional . . .	26
15	Montagem com configurações de pernas traseiras e dianteiras assimétricas	26
16	Configuração com replicação de pernas	27
17	Configuração de robô hexápode	27
18	Configuração de juntas e ligamentos em montagem alternativa	28
19	Montagem da plataforma em configuração alternativa	28
20	Interesse na atividade como extensão	30
21	Opinião sobre impacto na formação profissional	30
22	Mudança na motivação com a graduação caso atividade fosse ofertada	31

LISTA DE ABREVIATURAS

MQTT *Message Queuing Telemetry Transport*

UTF-8 *UCS Transformation Format 8*

JSON *JavaScript Object Notation*

MDF *Medium Density Fiberboard*

GHz Gigahertz

MP Megapixel

LED *Light Emitting Diode*

SD *Secure Digital*

GPIO *General Purpose Input/Output*

PWM *Pulse-Width Modulation*

USB *Universal Serial Bus*

I2C *Inter-Integrated Circuit*

DC *Direct Current*

UBEC *Universal Battery Eliminator Circuit*

HTTP *HyperText Transfer Protocol*

1 INTRODUÇÃO

A taxa de evasão em cursos de ensino superior pode representar um indicador de que a graduação não correspondeu às expectativas iniciais dos discentes. Outra possibilidade que se faz presente é aquela em que a graduação condiz com a idealização do aluno. Entretanto, surge desmotivação por parte do mesmo em decorrência da escassez de experiências proporcionadas pela instituição de ensino focadas em atividades práticas (CASSIANO et al., 2021).

No Brasil, mais da metade dos estudantes de engenharia abandonam o curso antes da conclusão (INDÚSTRIA, 2018). A Seção 3.1.1 apresenta uma pesquisa realizada com discentes de engenharia da Universidade do Rio Grande do Sul, na qual constatou-se que 48,9% dos estudantes sentiam-se desmotivados com o curso de graduação escolhido.

Levando esses pontos em consideração, apresenta-se uma proposta de atividade em que estudantes de etapas iniciais de cursos superiores de graduação em engenharia têm a possibilidade de interagir com uma plataforma multidisciplinar. A atividade tem como objetivo promover interesse nos estudantes sobre os diferentes sistemas incorporados ao projeto, assim como aumentar a motivação em relação à graduação.

A plataforma utilizada, por sua vez, foi desenvolvida como um robô de locomoção por pernas e tem seu funcionamento dependente da correta integração de seus sistemas. O projeto também apresentou capacidade de comando remoto. Além disso, objetivou-se que o robô deveria ser capaz de ser reconfigurável e modular, assumindo diferentes configurações de juntas e ligamentos. Para isso, faz-se uso das mesmas peças, sendo necessárias apenas alterações na montagem e eventual reescrita de rotinas de programação.

Este trabalho apresentou dois objetivos principais: primeiramente, que o projeto desenvolvido despertasse a motivação dos estudantes, seguindo os fatores de projeto propostos; e, em segundo lugar, que o robô projetado fosse funcional e apresentasse o comportamento esperado.

Apresenta-se uma breve revisão da literatura, seguida das etapas do projeto realizado. Na sequência, são apresentados os resultados observados e por fim as conclusões obtidas.

2 REVISÃO DA LITERATURA

A seguir é apresentada uma breve revisão da literatura, contemplando robótica educacional e robôs de movimentação por pernas.

2.1 Robótica Educacional

A robótica pedagógica ou robótica educacional consiste no ambiente de aprendizado em que o aluno é apresentado à montagem, automação e controle de dispositivos mecânicos que podem ser controlados pelo computador (SILVA, 2009). A utilização da robótica faz o aluno buscar conceitos em diferentes disciplinas para chegar à resolução de problemas, proporcionando uma vivência prática do método científico, através da construção de protótipos (GUIMARÃES JÚNIOR, 2015).

Como a robótica educacional envolve uma visão mais abrangente do que apenas uma área de conhecimento, também são desenvolvidos a autonomia, a capacidade de pensar em múltiplas alternativas para um mesmo problema, competências associadas ao pensamento lógico, organização e planejamento de projetos envolvendo robôs e integração de conceitos de diferentes áreas (SILVA, 2009).

Para o desenvolvimento de robótica educacional, podem ser utilizados *kits* de robótica, facilmente encontrados no mercado. Esses *kits* podem ser focados em atividades de Ensino Fundamental ou Médio, em competições de robótica ou montagem de robôs móveis inteligentes, visando o público de Ensino Superior ou pós-graduação (SILVA, 2009). Como exemplo, um dos *kits* de robótica utilizados popularmente é o Lego Mindstorms, mostrado na Figura 1.

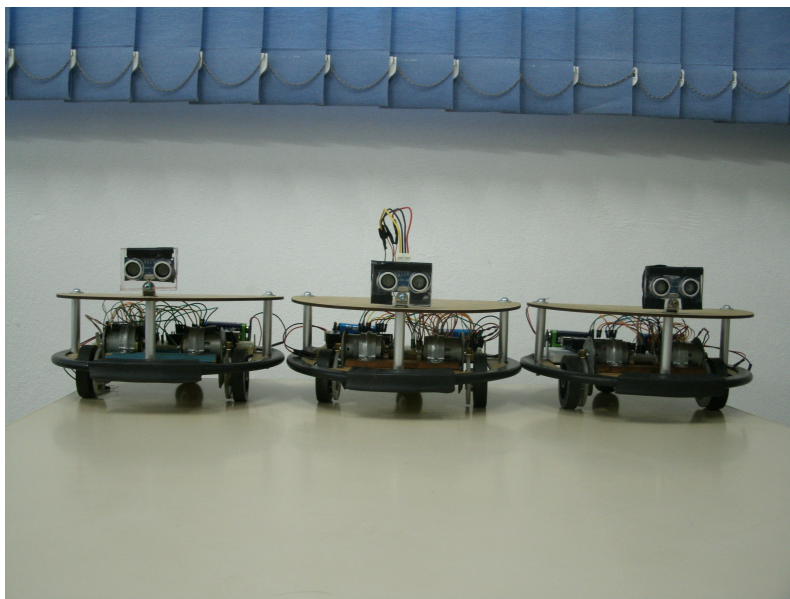
Figura 1: Robôs montados com o kit Lego Mindstorms



Fonte: Adaptado de (LEGO, 2021)

Outra alternativa para o desenvolvimento de atividades de robótica pedagógica é através de materiais reciclados ou sucatas. Fazendo o desmonte de equipamentos eletroeletrônicos obsoletos que seriam descartados, é possível obter tanto componentes eletrônicos - como sensores, transistores e reguladores de tensão - quanto elementos mecânicos e eletromecânicos - como motores, roldanas e engrenagens (SILVA, 2009). Um exemplo dessa metarreciclagem é o projeto EduBOT-v0.1, em que utilizam-se equipamentos descartados para realizar o desenvolvimento robótico, incentivando a criatividade sustentável na era digital (GUIMARÃES JÚNIOR, 2015). A Figura 2 mostra alguns robôs desse projeto.

Figura 2: Plataformas do projeto EduBOT utilizando metarreciclagem



Fonte: (EDUBOT, 2021)

2.2 Robôs com Locomoção por Pernas

Robôs com pernas se assemelham a animais e insetos, já que possuem ligamentos mecânicos para movimentação em superfícies irregulares (KEBRITCHI; HAVASHINEZHADIAN; ROSTAMI, 2018). O uso de robôs com pernas é amplamente utilizado em pesquisas de engenharia devido a esses serem muito mais versáteis que outros tipos de robôs (DELGADO ESTEBAN et al., 2016). Possíveis aplicações incluem missões de resgate em catástrofes, exploração espacial, operações de mineração e desarmamento de explosivos (DELGADO ESTEBAN et al., 2016).

Robôs hexápodes apresentam ótimo desempenho em áreas perigosas e instáveis, já que possuem maior capacidade de adaptação a terrenos irregulares e fazem contato com o solo por meio do ponto de contato de cada perna (CHENG, 2020). Robôs quadrúpedes, por sua vez, são escolhidos devido à sua estrutura menos complexa, levando a uma maior estabilidade durante as sequências de caminhada implementadas (A.RAHEEM; FLAYYIH, 2017). O desenvolvimento de robôs bípedes capazes de movimentação natural, rápida e versátil, enquanto mantêm alta eficiência energética, ainda é desafiador na questão de mecanismos, uso de atuadores e sistema de controle (MING et al., 2014).

O projeto de robôs com movimentação por pernas é complexo e apresenta alguns

desafios, como a navegação sobre obstáculos com ligamentos que sejam energeticamente (KEBRITCHI; HAVASHINEZHADIAN; ROSTAMI, 2018). A eficiência energética, assim como alta capacidade de torque dos atuadores, faz-se necessária para implementação de movimentações rápidas e naturais (MING et al., 2014).

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentadas as etapas de projeto para a plataforma robótica educacional desenvolvida.

3.1 Projeto Informacional

Nas próximas subseções é apresentada a etapa informacional do projeto. Realizaram-se uma análise do problema e uma breve pesquisa de mercado sobre produtos disponíveis para mitigar a situação identificada.

3.1.1 Análise do Problema

Realizou-se uma pesquisa com o objetivo de coletar informações sobre a motivação dos alunos com o curso de graduação escolhido. Os respondentes foram discentes de cursos superiores de Engenharia de Controle e Automação, Engenharia da Computação e Engenharia Elétrica da Universidade Federal do Rio Grande do Sul.

Os alunos também foram indagados sobre o aumento no interesse acadêmico com a possibilidade de uma atividade de robótica com foco prático, envolvendo o desenho de peças em *software* e programação de rotinas em embarcados. As informações coletadas através dessa pesquisa são mostradas na Figura 3

Figura 3: Respostas de discentes quanto a motivação com o curso e com atividades práticas



Fonte: Autor.

Os dados coletados indicaram que a atividade seria bem recebida por discentes e aumentaria a motivação dos mesmos.

3.1.2 Pesquisa de Mercado

Consultaram-se diferentes modelos de conjuntos de robótica educacional móvel disponíveis no mercado, assim como robôs vendidos comercialmente, sem foco na educação, mas possuindo estrutura de locomoção por pernas. Consideraram-se apenas produtos que permitiam ao usuário métodos de programação de rotinas no robô.

Os valores dos produtos foram adquiridos diretamente dos endereços das lojas virtuais dos fabricantes. No caso de produtos importados, não foram consideradas taxas de entrega e impostos sob importação, e a cotação de dólares americanos utilizada foi de R\$ 5,42, valor consultado diretamente de (BRASIL, 2021) durante o período de desenvolvimento deste trabalho. O resultado dessa consulta é apresentado na Tabela 1.

Tabela 1: Robôs móveis disponíveis comercialmente

Produto	Tipo	Valor (R\$)
EasyBotDS2	Movimentação por esteiras	499,00 (LIVRE, 2021c)
Kit Arduino Start Robô Controlado por App	Movimentação por rodas	199,00 (ÔMEGA, 2021a)
Kit Robótico Quadrúpede Adeept	Movimentação por pernas	436,60 (ADEEPT, 2021b)
Kit Robótico Hexápode Adeept	Movimentação por pernas	540,30(ADEEPT, 2021a)
Kit Robô Quadrúpede SunFounder V2.0	Movimentação por pernas	763,99 (SUNFOUNDER, 2021)
LOBOT Alienbot Micro	Movimentação por pernas	1554,32 (BANGGOOD, 2021)

Fonte: Autor.

3.2 Projeto Conceitual

Nesta seção são definidos os requisitos pertinentes ao desenvolvimento do projeto. Isso tem como objetivo guiar as etapas posteriores, ao definir fatores de projeto que devem ser seguidos.

3.2.1 Fatores de Projeto

Com base na pesquisa de mercado realizada na Seção 3.1.2, foi estabelecido que um dos fatores de projeto seria o custo, não devendo ultrapassar R\$ 500,00. Também foi possível observar que todos os robôs de movimentação por pernas consultados realizavam a sua locomoção por meio de servomotores elétricos. Assim, este projeto fez uso desse tipo de atuador eletromecânico.

Definiu-se também que a dimensão da estrutura principal do robô, desconsiderando os ligamentos das pernas, seria inicialmente estimada como um quadrado de quinze centímetros de lado. Essa estimativa foi decorrente da observação dos robôs na pesquisa de mercado e também levou em consideração o método de manufatura escolhido pelo autor, detalhado na seção 3.3.1.

Também foi definido que essa estrutura principal, assim como peças auxiliares de sustentação e peças que compõem os ligamentos das pernas, deveriam apresentar espessura de três milímetros. Essa definição teve como objetivo possibilitar o uso de variados métodos de manufatura, como corte a laser de chapas de *MDF* e acrílico ou usinagem de chapas de alumínio.

Outro fator de projeto foi a modularidade do robô. Deveria ser possível desenhar peças novas para a estrutura e integrá-las ao projeto sem a necessidade de alteração das demais peças. Além disso, as peças de ligamento das pernas deveriam ser reconfiguráveis,

permitindo que uma mesma junta tivesse sua rotação alterada entre eixos ortogonais. Isso teve o objetivo de reduzir o gasto com material e o tempo gasto com a manufatura de peças.

A plataforma educacional desenvolvida também deveria ser capaz de ser comandada remotamente por meio da implementação de protocolos de comunicação que utilizem-se de WiFi ou Bluetooth. Essa decisão teve como intuito possibilitar aos discentes adquirir experiência com essas implementações e interfaces.

3.3 Projeto Preliminar

Esta seção apresenta a etapa preliminar do projeto. É realizada a seleção de materiais e componentes utilizados em uma abordagem inicial, indo ao encontro dos fatores de projeto definidos.

3.3.1 Seleção do Material

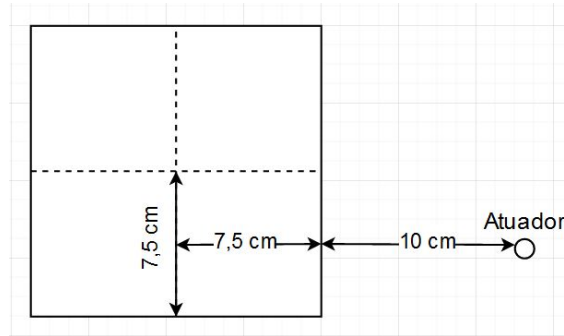
Visando a prototipagem rápida das peças desenvolvidas para o robô, o material utilizado foi PLA - ácido polilático, e o método de manufatura escolhido foi impressão 3D utilizando uma impressora *Ender V2 Pro*, devido à disponibilidade ao autor. Essa impressora apresenta capacidade para imprimir peças de até 220 x 220 mm, e mostrou-se adequada para a manufatura do robô, de acordo com a estimativa feita da estrutura principal.

Ressalta-se que esse material foi utilizado apenas para fase de prototipagem do projeto, considerando que a utilização de peças em *MDF* manufaturadas por corte a laser vêm ao encontro ao fator de projeto de redução de custo. Para a seleção dos atuadores, considerou-se uma estrutura de PLA, que possui densidade de $1,24 \text{ g/cm}^3$ (3DLAB, 2021), enquanto *MDF* possui densidade entre 0,5 a 1 g/cm^3 (CAMEO, 2021).

3.3.2 Seleção dos Atuadores

De modo a realizar a escolha dos atuadores, considerou-se que a estrutura principal da plataforma seria composta de duas peças principais, nas porções superior e inferior. Também fez-se uso de suportes laterais para sustentação, esses de massa desprezível em comparação com as peças principais, especificadas na Seção 3.2.1.

Estimou-se uma situação preliminar em que as duas peças principais seriam sustentadas por quatro atuadores, cada um posicionado em um ligamento de dez centímetros de comprimento. O encaixe desses ligamentos foi considerado na extremidade da estrutura principal. Realizou-se com essas considerações uma análise do torque necessário em cada servomotor para a sustentação da estrutura. A configuração de montagem dos atuadores escolhida para esta análise é representada na Figura 4.

Figura 4: Configuração utilizada na análise de torque

Fonte: Autor.

Considerando a densidade do PLA, e que um quarto de uma das peças principais da estrutura possui uma área de $56,25 \text{ cm}^2$, como mostrado na Figura 4, com espessura de três milímetros, tem-se uma massa de 20,925 gramas para cada peça e de 41,85 g para o conjunto das duas peças, como mostrado na Equação 1.

$$m_{estrutura} = 1,24 \text{ g/cm}^3 \cdot 56,25 \text{ cm}^2 \cdot 0,3 \text{ cm} \cdot 2 = 41,85 \text{ g} \quad (1)$$

Para fins de cálculo, essa massa é considerada aplicada no ponto central do quadrante de atuação de um servomotor, à uma distância de 13,75 cm. O torque resultante é calculado na Equação 2, usando uma aproximação da gravidade como $9,8 \text{ m/s}^2$.

$$T = 0,4185 \text{ kg} \cdot 9,8 \text{ m/s}^2 \cdot 0,1375 \text{ m} = 0,056392875 \text{ N/m} = 0,5750 \text{ kgf} \cdot \text{cm} \quad (2)$$

Realizou-se uma pesquisa em fornecedores, e foram coletados dados referentes aos servomotores, mostrados na Tabela 2. Esses atuadores foram selecionados devido a estarem disponíveis para pronta-entrega em todos os fornecedores consultados. O torque de partida refere-se a situação em que o servomotor está alimentado com tensão de 4,8 V. Foram considerados atuadores de ângulo de rotação de 0 até 180 graus, intervalo julgado suficiente para movimentação das pernas do robô.

Tabela 2: Servomotores elétricos disponíveis comercialmente

Servomotor	Torque de Partida (kgf.cm)	Massa (g)	Valor (R\$)
SG90	1,2	9	20,80 (ELETRÔNICA, 2021a)
SG92R	2,5	9	25,90 (ÔMEGA, 2021b)
MG90	1,8	13,4	30,95 (ELETRÔNICA, 2021a)
MG946R	10,5	55	49,90 (ELETROGATE, 2021d)
MG996R	9,4	55	59,90 (ELETROGATE, 2021f)
MG995	13	69	59,90 (ELETROGATE, 2021e)

Fonte: Autor.

3.3.3 Seleção do Microcontrolador

De modo a possibilitar que a plataforma fosse comandada remotamente, deve ser escolhido um microcontrolador que possua funcionalidades de comunicação, seja via WiFi, rádio-frequência ou Bluetooth. Foi realizada uma pesquisa entre diferentes fornecedores considerando microcontroladores que se encaixavam no requisito proposto, mostrados na Tabela 3. Já que os atuadores escolhidos para o projeto tinham sua movimentação determinada por sinal *PWM - Pulse Width Modulation* (Modulação por Largura de Pulso), está evidenciado o número de portas digitais que oferecem suporte à essa funcionalidade de cada microcontrolador.

Tabela 3: *Microcontroladores com suporte à conexão remota disponíveis comercialmente*

Microcontrolador	Portas com suporte a <i>PWM</i>	Interface USB serial	Câmera	Valor (R\$)
ESP-12F	11	Micro-USB	Não	59,90 (FLOP, 2021a)
NodeMCU	38	Micro-USB	Não	63,00 (LIVRE, 2021b)
ESP32-CAM	10	Não	2 MP	50,00 (LIVRE, 2021a)
Wemos D1	10	Micro-USB	Não	78,84(ELETRÔNICA, 2021b)

Fonte: Autor.

Dentre os modelos disponíveis comercialmente, escolheu-se o ESP32-CAM, devido ao fato de possuir uma câmera integrada de 2 Megapixels, possibilitando a implementação de rotinas de visão computacional em trabalhos futuros.

Esse microcontrolador, entretanto, possui apenas 10 pinos de entrada e saída que suportam *PWM*, o que limita o número de atuadores e sensores que podem ser utilizados simultaneamente. Com isso, decidiu-se fazer uso de um módulo expensor de portas *PWM* PCA9685. Esse componente comunica-se com o microcontrolador pelo protocolo *I2C - Inter Integrated Circuit* (Circuito Intra-Integrado) e possibilitou o uso de 16 portas *PWM* extras, permitindo ao usuário um maior número de configurações de juntas possíveis.

3.3.4 Seleção de Elementos Elétricos de Potência

Considerou-se o acionamento de apenas quatro servos por vez de uma posição estática até a movimentação, apresentando seu valor de corrente de partida. Devido à ausência dessa informação em consulta com o fabricante, consideraram-se dados experimentais de um fornecedor sobre esse valor de corrente para o servomotor SG92R, apresentando até 650 mA de corrente de partida (SUPPLIES, 2021). Também foi considerado que em condição de movimentação nominal, o atuador necessita de até 300 mA de corrente. Com isso, considerando o acionamento de quatro servos, estimou-se um consumo momentâneo de corrente de 2,6 A. Após a fase de partida, necessita-se de 1,2 A para os atuadores.

Devido ao caráter educacional do projeto, assumiu-se a premissa de que quando utilizado dentro de instituições acadêmicas, seria possível fazer uso de fontes de bancada para a alimentação da plataforma. Também considerou-se o uso de duas baterias recarregáveis modelo 1850 de 3,7 V e 2200 mAh em série, tanto para a fase de prototipagem do robô quanto para permitir uma maior mobilidade em trabalhos futuros. Entretanto, a massa dessas baterias não foi considerada na análise de torque realizada, implicando que a fonte de alimentação deveria ser posicionada externamente à plataforma e cabeada, ou então a análise de torque deveria ser refeita.

Com o uso de baterias na etapa de prototipagem, decidiu-se segregar o fornecimento

de energia para o robô em duas parcelas. Uma delas levaria energia para os servomotores, denominada parte de potência, e a outra alimentaria o microcontrolador e outros sensores, denominada parte lógica. O microcontrolador escolhido opera em 5 V e faz uso de 180 mA em operação nominal (STUDIO, 2021). Os servomotores são acionados pelo módulo de extensão *PWM* apresentado na seção 3.3.3, que aceita de 5 V a 6 V de entrada de tensão. Como a fonte de alimentação considerada apresentaria 7,4 V, fez-se necessário o uso de elementos de conversão de tensão elétrica.

Na Tabela 4, são apresentados os resultados de uma pesquisa de mercado realizada sobre conversores de tensão que operam nas especificações apresentadas.

Tabela 4: *Conversores de tensão disponíveis comercialmente*

Conversor	Corrente Nominal (A)	Corrente máxima (A)	Valor (R\$)
7805	1	2,2	2,50 (ROBOCORE, 2021)
XL4005	2,5	5	19,90 (ELETROGATE, 2021c)
LM2596	2	3	13,90 (FLOP, 2021b)
AMS1117	1,1	1,5	8,90 (ELETROGATE, 2021b)
UBEC	3	5	33,00 (FLIGHTMANIA, 2021)
LM317	1,5	2,2	12,90 (ELETROGATE, 2021a)

Fonte: Autor.

Para o acionamento dos servomotores, optou-se pela utilização do Conversor UBEC, por apresentar a maior corrente nominal e de partida entre os componentes considerados, como pode ser observado na Tabela 4. Embora apresente o maior custo dentre as opções, considerou-se de maior relevância a possibilidade de aumentar o número de atuadores da plataforma em trabalhos futuros.

Para a parte lógica, considerou-se a utilização do conversor LM2596 devido à pronta-entrega por parte do fornecedor, eficiência de até 92% (FLOP, 2021b) e à sua corrente nominal, que permite a implementação de módulos de sensores adicionais em trabalhos futuros.

3.3.5 Seleção do Protocolo de Comunicação

A interface de comunicação com o robô deveria ser de fácil implementação, assumindo o pressuposto que o mesmo seria utilizado em atividades introdutórias a protocolos de comunicação sem fio.

O protocolo MQTT - *Message Queuing Telemetry Transport* (Transporte de Mensagens por Telemetria Sequenciada) é considerado o protocolo padrão de IoT - *Internet of things* (Internet das coisas). Foi desenvolvido inicialmente pela empresa IBM nos anos 90. Sua aplicação inicial consistia em realizar a comunicação entre satélites e sensores em alto-mar. Por se tratar de um protocolo assíncrono, permite aos dispositivos conectados estarem separados tanto pelo espaço quanto pelo tempo. Com isso, é um protocolo escalável em ambientes de rede que não são confiáveis. (YUAN, 2017)

Este protocolo oferece suporte a várias plataformas e linguagens de programação, sendo amplamente utilizado em aplicações *Android* (TANG et al., 2013). Voltado para aplicações com baixa largura de banda e alta latência, os clientes estabelecem conexão com um *broker* (gerenciador de comunicação) e então trocam mensagens por meio de tópicos (ADOLPHS, 2020), que são *strings* no formato UTF-8.

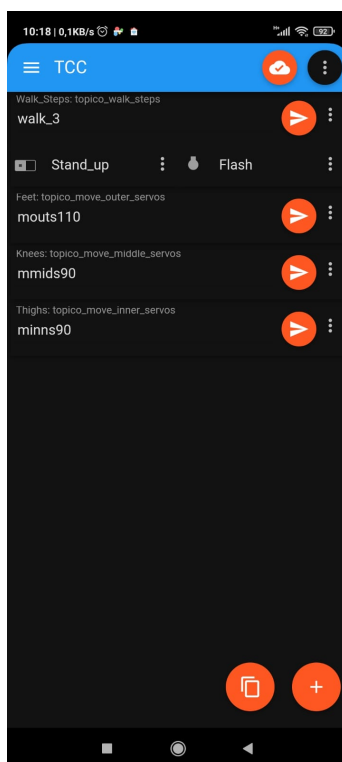
É possível realizar uma implementação em que as mensagens transmitidas em um determinado tópico são no formato JSON (*JavaScript Object Notation* - Notação de Objetos Javascript). Assim, é possível realizar o envio de múltiplos pares de dados estruturados em coleções de pares/valor por meio de uma única mensagem de texto (TEIXEIRA, 2019).

O protocolo HTTP - *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto) é utilizado em bibliotecas disponíveis para o microcontrolador escolhido. As rotinas que fazem uso desse protocolo apresentam uma transmissão de imagens com baixa latência do microcontrolador para um endereço virtual pré-determinado. Analisou-se a possibilidade de adaptação dessas rotinas para incorporar comandos ao robô.

Observou-se que o reuso da biblioteca já disponível em HTTP era possível, entretanto dificultava a implementação de novos comandos, e julgou-se que a complexidade envolvida na adequação de rotinas poderia ser evitada com o uso do protocolo MQTT. Assim, durante a fase de prototipagem, utilizou-se MQTT como o protocolo de comunicação. Embora a transmissão de imagens tenha apresentado um desempenho inferior com o uso desse protocolo, essa funcionalidade foi considerada fora do escopo deste trabalho.

Foi utilizado o aplicativo IoT MQTT Panel, do desenvolvedor Rahul Kandul, disponível para Android, para enviar comandos a plataforma. Essa aplicação foi escolhida por ser gratuita, de fácil configuração e, após testes, ter seu funcionamento comprovado no envio de mensagens. Outros aplicativos poderiam ser utilizados para comunicação com o robô de acordo com a preferência e sistema operacional do usuário, desde que operassem de acordo com o protocolo selecionado. A interface do aplicativo IoT MQTT Panel é mostrada na Figura 5.

Figura 5: Interface do aplicativo utilizado para comunicação via protocolo MQTT



Fonte: Autor.

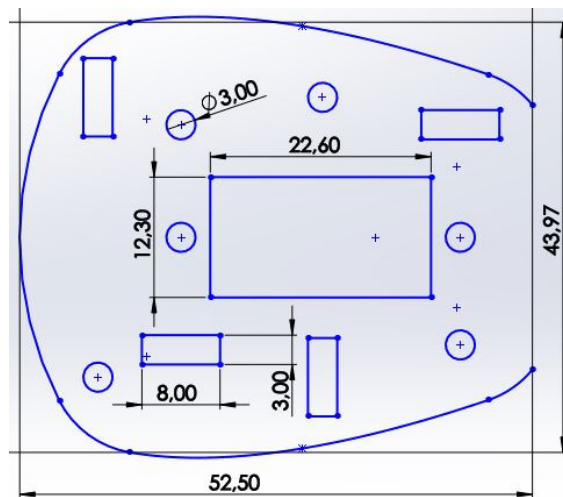
3.4 Projeto Detalhado

Na etapa de projeto detalhado, realizam-se o desenho da estrutura do robô e a montagem da plataforma na configuração escolhida para estudo de caso. Também determinam-se as expressões de cinemática inversa para a configuração escolhida.

3.4.1 Desenho de Padrão de Encaixe Modular

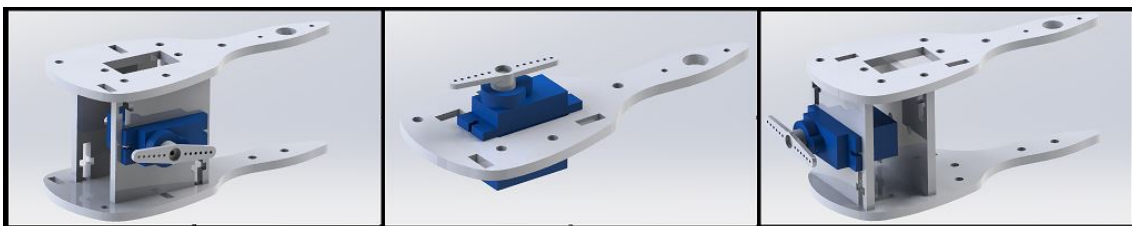
Realizou-se a concepção de um padrão de encaixe, mostrado na Figura 6, com o auxílio do programa *SolidWorks*. Isso foi realizado considerando as dimensões do servomotor escolhido, a fim de proporcionar ao mesmo a possibilidade de montagem em três eixos ortogonais, como é exemplificado na Figura 7. Essa possibilidade visou atender principalmente o fator de projeto de permitir à plataforma ser reconfigurável.

Figura 6: Padrão de encaixe modular desenvolvido



Fonte: Autor.

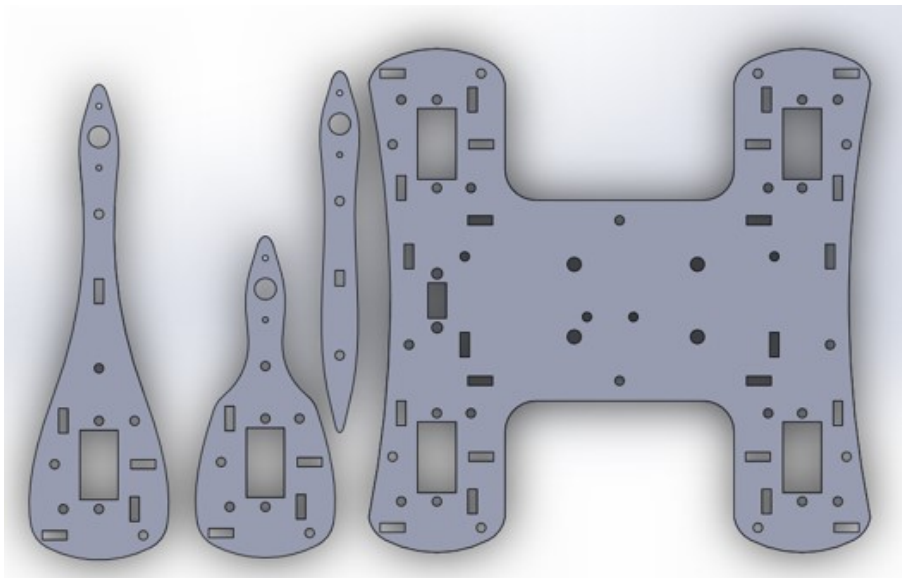
Figura 7: Diferentes possibilidades de posicionamento de um atuador



Fonte: Autor.

Ao desenvolver um padrão de encaixe para este fim, o fator de projeto de modularidade da plataforma também foi contemplado, já que este padrão pôde ser incorporado em diferentes peças. A Figura 8 mostra as diferentes peças desenvolvidas para a estrutura principal e para os ligamentos do robô, onde pode ser observado o uso do padrão de encaixe.

Figura 8: Peças desenhadas para a plataforma



Fonte: Autor.

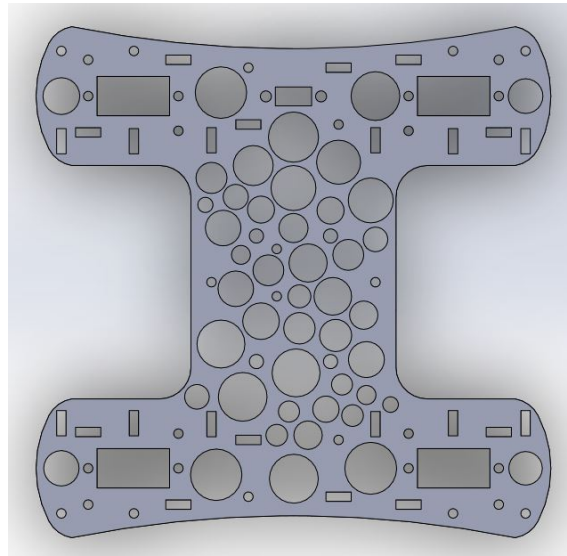
É possível visualizar na Figura 8 as principais peças desenvolvidas para o robô. Da esquerda pra direita, têm-se: um ligamento de dez centímetros, um ligamento de cinco centímetros, um efetuador de dez centímetros e a peça de estrutura principal da plataforma, referenciada anteriormente na Seção 3.3.2.

Essa peça principal foi assumida na Seção 3.3.2 como tendo suporte à montagem de quatro servomotores, espessura de três milímetros, e largura e comprimento de quinze centímetros. Como é possível perceber na Figura 8, essa peça foi desenhada diferentemente dessa estimativa com o objetivo de redução de material. Esse componente também apresentou furos projetados para a fixação dos componentes eletrônicos.

Também é possível perceber que os ligamentos, por serem projetados com o padrão de encaixe, não estavam completamente adequados à análise de torque feita inicialmente, em que sua massa foi desconsiderada. Entretanto, considerou-se que a redução de material da peça de estrutura principal mitigou esta inadequação para configurações do robô que não façam uso de um número expressivo de peças de ligamento.

Com o objetivo de realizar uma redução ainda maior de massa da peça de estrutura principal, o desenho da mesma foi alterado e é mostrado na Figura 9. Esse novo desenho de peça apresentou uma redução de um quarto de massa, enquanto ainda mantendo integridade estrutural julgada suficiente para as possíveis aplicações da plataforma. As peças utilizadas no desenvolvimento deste projeto podem ser encontradas em github.com/Leo13Z/Plataforma_Educacional_de_Desenvolvimento_Robotica.

Figura 9: *Desenho da peça de estrutura principal visando a redução de massa*

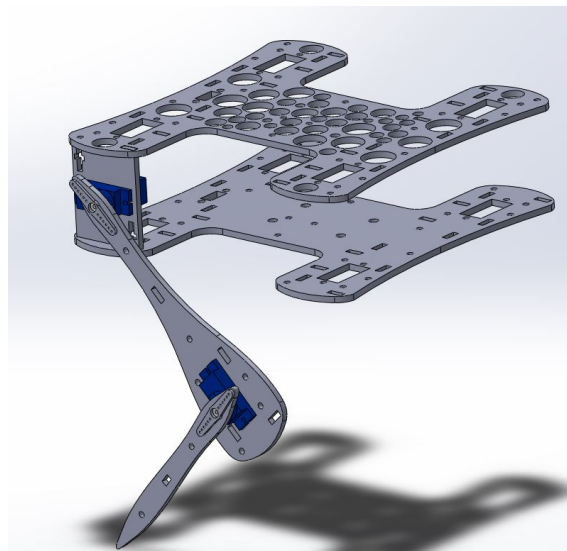


Fonte: Autor.

3.4.2 Estudo de Caso

As peças projetadas foram impressas, e realizou-se a montagem da estrutura do robô. Tendo em vista que a plataforma deveria ser reconfigurável em relação à orientação das juntas e ligamentos, escolheu-se arbitrariamente a configuração mostrada na Figura 10 para todas as pernas do robô. Essa configuração possuía dois graus de liberdade para cada perna, com as juntas montadas na mesma orientação e ligamentos e efetuator de dez centímetros.

Figura 10: *Configuração de juntas e ligamentos escolhida para estudo de caso*

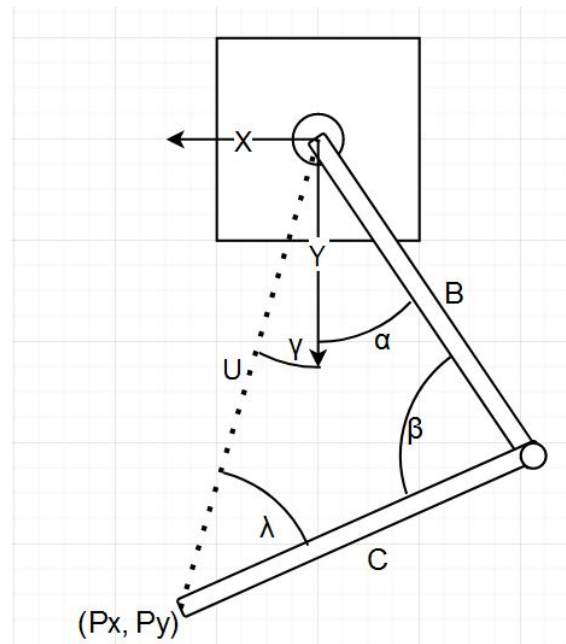


Fonte: Autor.

3.4.3 Cálculo da Cinemática Inversa

Observou-se a necessidade de um comando preciso da posição da extremidade do efetuator para implementação de movimentações complexas. Para a determinação da expressão de cinemática inversa da configuração escolhida para estudo de caso, consideraram-se o sistema de coordenadas e a pose mostrados na Figura 11. O método de determinação de cinemática inversa escolhido foi o método geométrico.

Figura 11: Pose considerada para o cálculo da cinemática inversa



Fonte: Autor.

Nesta configuração, B e C representavam as dimensões do ligamento e efetuator utilizados, respectivamente, e Px e Py demarcavam a posição espacial da extremidade do atuador. Analisando a Figura 11, foi possível determinar a distância do ponto desejado até a origem do sistema de coordenadas proposto de acordo com a Equação 3.

$$U = \sqrt{Px^2 + Py^2} \quad (3)$$

Foi realizado o cálculo de γ , como mostrado na Equação 4.

$$\gamma = \arctan \frac{Px}{Py} \quad (4)$$

Fez-se uso da lei dos cossenos para realizar a determinação do ângulo composto de $\gamma + \alpha$, como mostrado na Equação 5.

$$\gamma + \alpha = \arccos \frac{B^2 + U^2 - C^2}{2 \cdot B \cdot U} \quad (5)$$

Com isso, determinou-se o valor de α , mostrado na Equação 6.

$$\alpha = (\gamma + \alpha) - \gamma \quad (6)$$

O valor do ângulo do atuador seguinte foi calculado fazendo uso da lei dos senos, como mostrado na Equação 7.

$$\beta = \arcsin \frac{U \cdot \sin(\gamma + \alpha)}{C} \quad (7)$$

Realizou-se ajustes no cálculo desses valores na rotina de programação implementada, com o objetivo de operar sempre no primeiro e segundo quadrante. Isto foi devido aos servomotores escolhidos possuírem movimentação de 0 a 180 graus. Também foram consideradas condições de contorno e ajustes devido à montagem dos atuadores na estrutura. A parcela do código desenvolvida para a realização do cálculo da cinemática inversa é mostrada na Figura 12, e pode ser visualizada no contexto da rotina implementada no Apêndice B.

Figura 12: Função desenvolvida em C++ para o cálculo de cinemática inversa

```
void inverseKinematic(float x, float y, int* al, int* a2)
{
    double U = sqrt((x*x)+(y*y));
    double gamma = atan2(x, y);
    double gamma_com_alfa;
    if (((B*B)+(U*U)-(C*C))/(2*B*U) < 1)
        gamma_com_alfa = acos(((B*B)+(U*U)-(C*C))/(2*B*U));
    else
        gamma_com_alfa = acos(1);
    double alfa = gamma_com_alfa - gamma;
    double beta;
    if ((U*sin(gamma_com_alfa)/C) < 1)
        beta = asin(U*sin(gamma_com_alfa)/C);
    else
        beta = asin(1);
    if (U > sqrt((B*B)+(C*C)))
    {
        beta = (PI) - beta;
    }
    int anglemid = 90 - int(alfa * (180/PI));
    int angleout = int(beta * (180/PI));
    if ((anglemid>0) && (anglemid<180)){
        *al=anglemid;
    }
    if ((angleout>0) && (angleout<180)){
        *a2=angleout;
    }
}
```

Fonte: Autor.

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos em operação nominal da plataforma em uma configuração escolhida como estudo de caso. Também são apresentadas outras possibilidades de montagem do robô e resultados da pesquisa de aceitação discente realizada sobre a atividade proposta.

4.1 Operação Nominal

Realizou-se a integração dos componentes selecionados com a estrutura montada para o robô, na configuração de ligamentos e juntas exposta na Seção 3.4.2. A montagem finalizada do robô na configuração escolhida é mostrada na Figura 13. Verificou-se o envio de mensagens por um aparelho móvel utilizando a interface e protocolo de comunicação escolhidos, assim como o recebimento por parte do microcontrolador.

Observou-se que as extremidades dos efetadores posicionaram-se de acordo com a posição espacial comandada, validando a rotina de cinemática inversa implementada. Foi verificado que os atuadores utilizados não demonstraram problemas devido a excesso de carga, e foram capazes de realizar a sustentação e movimentação da plataforma.

Figura 13: Montagem da plataforma para estudo de caso



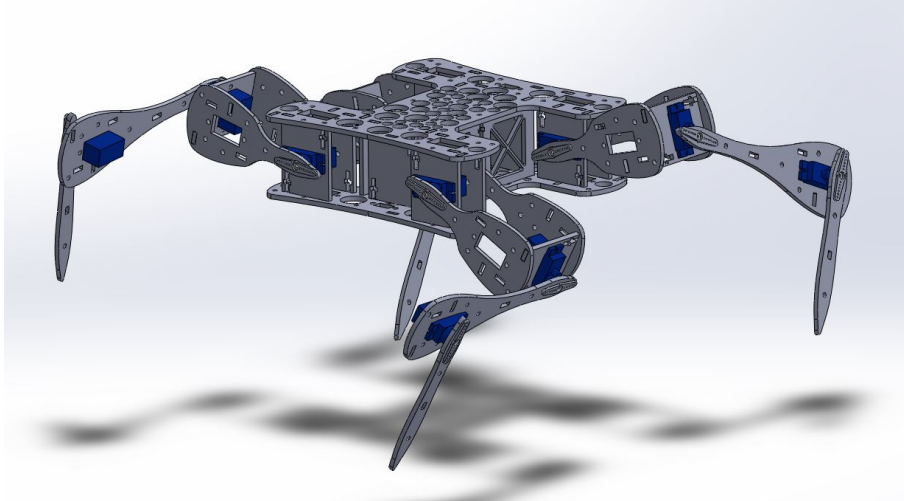
Fonte: Autor.

4.2 Validação de Reconfiguração

Com o objetivo de comprovar a capacidade de reconfiguração e modularidade do robô, foram propostas as montagens apresentadas a seguir. A configuração mostrada na Figura

14 consiste na adição de um grau de liberdade adicional em cada perna da montagem escolhida como estudo de caso, aumentando a capacidade de movimentação da plataforma.

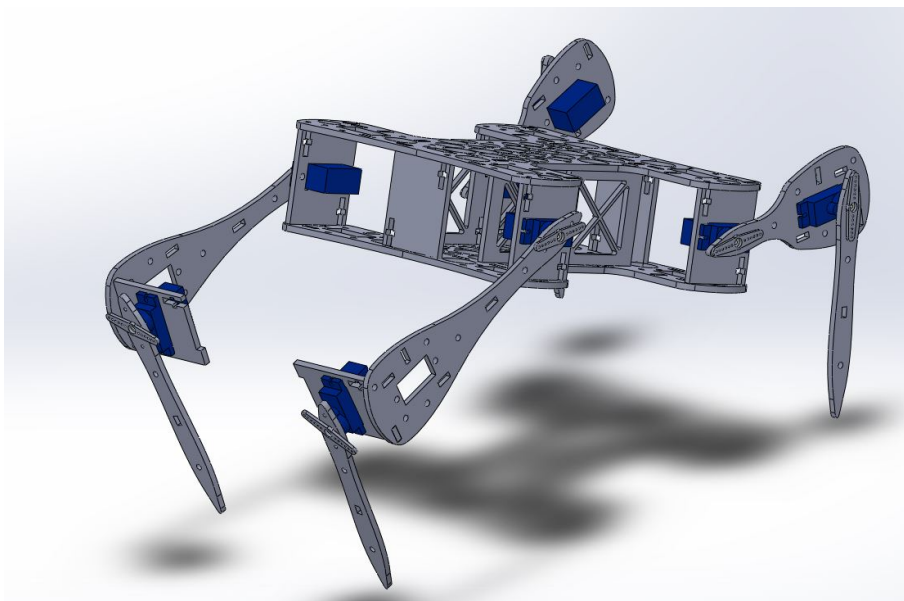
Figura 14: *Configuração de estudo de caso com grau de liberdade adicional*



Fonte: Autor.

A montagem apresentada na Figura 15 exemplifica a possibilidade de montagens assimétricas. Neste caso, as pernas dianteiras e traseiras da plataforma possuem configurações distintas tanto de juntas como de ligamentos. Tanto esta montagem quanto a mostrada na Figura 14 fazem uso de peças já desenvolvidas, comprovando a intercambialidade de peças entre configurações.

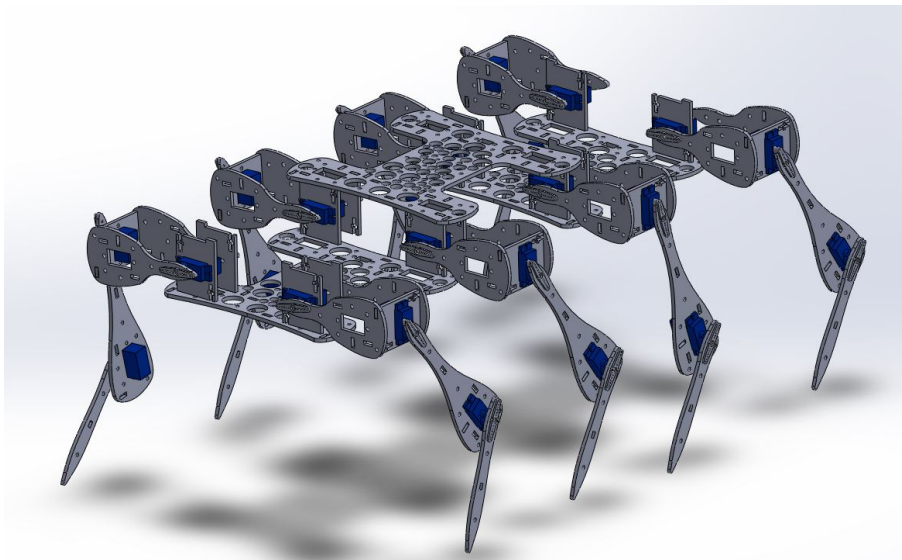
Figura 15: *Montagem com configurações de pernas traseiras e dianteiras assimétricas*



Fonte: Autor.

Na montagem mostrada na Figura 16, a configuração de pernas utilizada na proposta de montagem mostrada na Figura 14 é replicada fazendo uso de peças de estrutura principal adicionais, sendo um exemplo da modularidade do sistema. Essa é uma possibilidade para a utilização de um número maior de pernas, como exemplificado na figura.

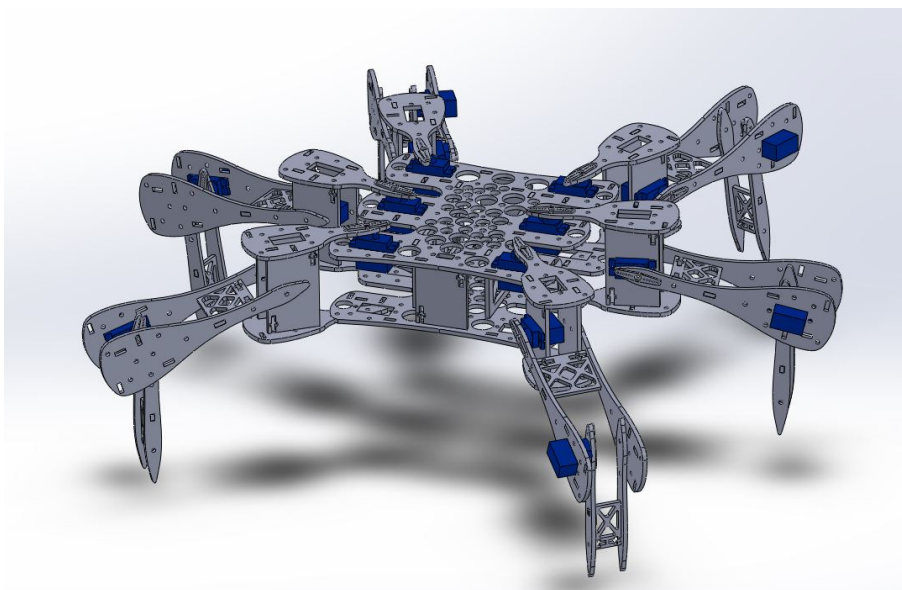
Figura 16: *Configuração com replicação de pernas*



Fonte: Autor.

Devido a modularidade do padrão de encaixe desenvolvido em si, o desenho de novas peças específicas para a configuração desejada também é uma possibilidade, apresentando pouca complexidade. A Figura 17 comprova esta afirmação, usando a peça principal da estrutura usada até o momento modificada para oferecer suporte a duas pernas adicionais, tornando a plataforma um robô hexápode.

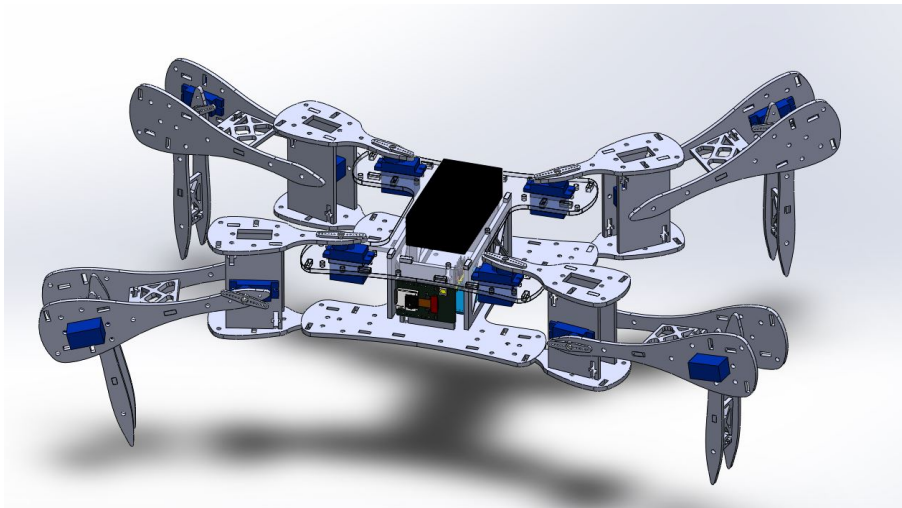
Figura 17: *Configuração de robô hexápode*



Fonte: Autor.

Com o objetivo de comprovar o processo de reconfiguração no protótipo desenvolvido, optou-se por realizar uma montagem com as juntas e ligamentos da plataforma configurados como mostrado na Figura 18. Observou-se que o tempo necessário para alterar entre as diferentes configurações de montagem foi de aproximadamente 1 hora.

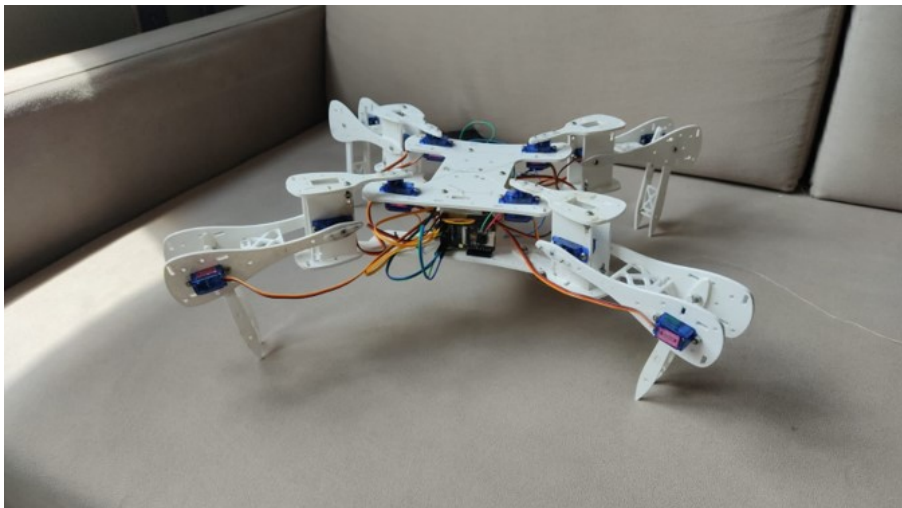
Figura 18: *Configuração de juntas e ligamentos em montagem alternativa*



Fonte: Autor.

O protótipo pode ser observado montado nessa configuração alternativa ao estudo de caso inicial na Figura 19. Devido à essa montagem apresentar orientações de juntas e tamanhos de ligamentos diferentes dos considerados no cálculo da cinemática inversa da Seção 3.4.3, comandaram-se diretamente os ângulos dos atuadores de forma remota em experimentos realizados. Embora o intuito principal seja a validação da possibilidade de reconfiguração da plataforma, optou-se por dobrar a massa dos ligamentos, como pode ser observado na Figura 19.

Figura 19: *Montagem da plataforma em configuração alternativa*



Fonte: Autor.

Foi possível observar que os atuadores não apresentaram o comportamento descrito na Seção 4.1, mostrando incapacidade de assumir o ângulo comandado. Realizou-se uma análise deste problema e constatou-se que a geração deste fato devia-se ao torque insuficiente dos servomotores na configuração montada. Com isso, notou-se que a análise de torque realizada apresentou resultados satisfatórios apenas quando foi possível desconsiderar a massa dos ligamentos. Assim, faz-se necessária em trabalhos futuros uma análise estrutural mais completa dos desenhos de peças de ligamento, no intuito da máxima redução de massa sem comprometimento da integridade estrutural necessária para o funcionamento do robô. Também recomenda-se uma análise dinâmica para cada configuração diferente utilizada, a fim de obter uma estimativa do torque necessário em cada servomotor.

Observou-se que com uma maior exigência de torque nos atuadores, a movimentação exigida das pernas ocasionou desgaste e quebra de componentes internos dos servomotores. Para trabalhos futuros, faz-se necessária uma análise da sequência de movimentação escolhida para cada perna, a fim de concluir se existe a possibilidade de mitigação deste problema por *software*. Outra alternativa que se faz presente é o uso de servomotores com elementos internos de transmissão metálicos, que apresentam maior durabilidade.

Considerando os pontos expostos, observou-se a possibilidade de substituição do servomotor escolhido pelo modelo MG995 futuramente, priorizando o aumento de possibilidades de configuração devido ao maior torque apresentado, como mostrado na Tabela 2. Essa substituição tem como consequência adaptação do padrão de encaixe modular para as dimensões desse servomotor. Além disso, há também a necessidade de análise de adequação dos elementos de conversão de tensão escolhidos considerando os requisitos de potência desse atuador.

Caso priorize-se a redução de custo em detrimento de reconfiguração, recomenda-se o uso do servomotor MG90, que possui as mesmas dimensões do servomotor utilizado, permitindo o uso do padrão de encaixe e estruturas físicas já existentes sem adequações. Este atuador apresenta um torque de partida levemente menor e aproximadamente a mesma corrente de partida dos atuadores selecionados na fase de prototipagem do projeto, descartando a necessidade de novos elementos elétricos de potência.

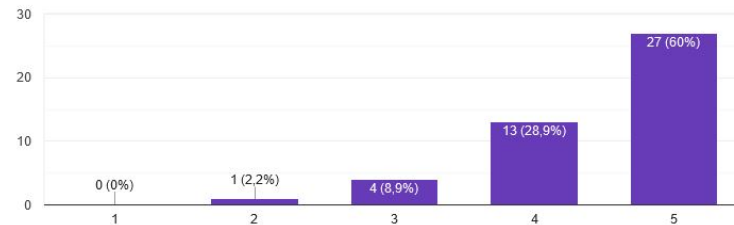
4.3 Aceitação Discente

Com o robô montado e funcional, o mesmo foi apresentado aos respondentes da pesquisa realizada, apresentada no Apêndice A. Quando questionados sobre escolher uma atividade de extensão envolvendo o projeto de novas peças para a plataforma e elaboração de rotinas de programação para movimentação, 88,9% dos alunos demonstraram uma resposta tendendo ao interesse na atividade, como é possível ver na Figura 20.

Figura 20: *Interesse na atividade como extensão*

Considerando a plataforma apresentada na imagem, dê uma nota de 1 a 5 pra sua motivação em escolher uma atividade de extensão que fosse desenvolver rotinas pra movimentação do robô de acordo com a configuração montada, assim como fazer novas "patas" e cortar em MDF ali na UFRGS pra testar diferentes tipos de movimentação.

45 respostas



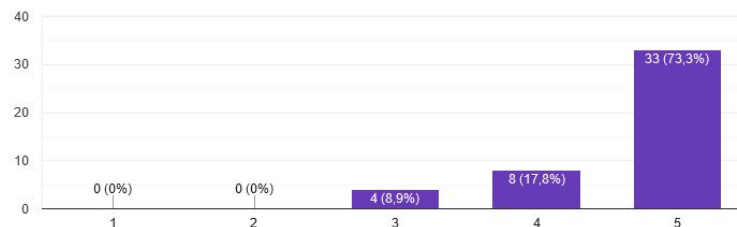
Fonte: Autor.

Além disso, 91,1% dos discentes consultados têm a opinião que uma atividade deste tipo, abordando na plataforma temas como protocolos de comunicação, cinemática inversa, desenho de peças em software e programação de microcontroladores, agregaria positivamente à sua formação profissional, como é mostrado na Figura 21.

Figura 21: *Opinião sobre impacto na formação profissional*

Considerando a atividade proposta anteriormente, envolvendo desenho em CAD, protocolos de comunicação, cinemática inversa e programação de microcontroladores, o quanto você acha que ela agregaria positivamente à sua formação como profissional?

45 respostas



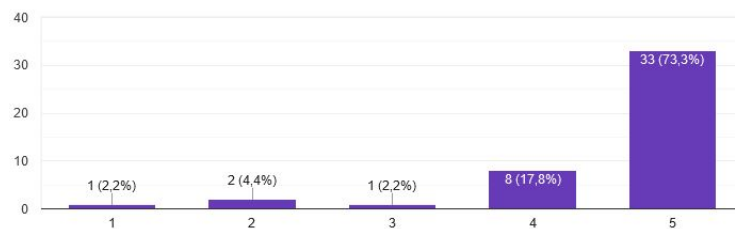
Fonte: Autor.

Quanto a motivação com o curso de formação, 91,1% dos respondentes afirmaram que se sentiriam mais motivados caso essa atividade proposta fosse realizada entre o terceiro e o sexto semestre da graduação, resultado esse mostrado na Figura 22.

Figura 22: Mudança na motivação com a graduação caso atividade fosse ofertada

Por fim, o quanto você considera que essa atividade teria um impacto positivo na sua motivação para com o curso caso fosse realizada nos semestres iniciais do curso, entre 3 e 6?

45 respostas



Fonte: Autor.

5 CONCLUSÕES

A plataforma robótica desenvolvida apresentou capacidades de reconfiguração e modularidade, apresentando possibilidade de adição de módulos eletrônicos que fazem uso do protocolo I2C. Apresentou-se também a possibilidade de incorporação de novas peças desenhadas, contanto que adequadas ao padrão de encaixe modular proposto. Também observou-se a implementação de um protocolo de comunicação remota, e todas as peças desenhadas apresentaram capacidade de manufatura por corte a laser de *MDF*, mantendo a acessibilidade.

Além disso, com os resultados da pesquisa realizada aos discentes, mostrou-se perceptível que as possíveis atividades de robótica educacional com o robô desenvolvido seriam bem recebidas, não só aumentando o ânimo dos estudantes com a graduação escolhida, mas também agregando valor à formação profissional dos alunos.

Para trabalhos futuros, recomendam-se uma análise estrutural mais aprofundada das peças desenvolvidas para redução de massa, assim como análises dinâmicas e determinação das expressões de cinemática inversa para diferentes configurações de montagem escolhidas.

REFERÊNCIAS

- 3DLAB. Você sabe quantos metros tem o filamento para impressão 3D? In. Disponível em: <<https://3dlab.com.br/quantos-metros-tem-o-filamento-para-impressao-3d/>>. Acesso em: 12 nov. 2021.
- A.RAHEEM, F.; FLAYYIH, M. K. Quadruped Robot Creeping Gait Stability Analysis and Optimization Using PSO. In: 2017 Second Al-Sadiq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA). [S.l.: s.n.], 2017. p. 79–84.
- ADEEPT. Adept Hexapod Spider Robot Kit for Arduino with Android APP and Python GUI, Spider Walking Crawling Robot, STEAM Robotics Kit with PDF Manual. In. Disponível em: <http://www.adeept.com/hexapod-spider-robot-v30_p0130.html>. Acesso em: 12 nov. 2021.
- ADEEPT. Adept Quadruped Robot Kit for Arduino with Infrared Remote Control and Python APP, Spider Walking Crawling Robot. In. Disponível em: <http://www.adeept.com/quadruped-robot_p0127.html>. Acesso em: 12 nov. 2021.
- ADOLPHS, R. S. *Projeto: Sistema de Irrigação Automático Supervisionado para Hortas*. 2020. f. 33. Trabalho de conclusão de curso – Universidade Federal do Rio Grande do Sul, Porto Alegre.
- BANGGOOD. LOBOT Alienbot Micro:bit Programmable Multifunctional PC/APP Control Smart RC Robot - assembled. In. Disponível em: <https://www.banggood.com/LOBOT-Alienbot-Micro-bit-Programmable-Multifunctional-PC-or-APP-Control-Smart-RC-Robot-p-1476889.html?rmmds=detail-topright-recommendation&cur_warehouse=CN&ID=6154613>. Acesso em: 12 nov. 2021.
- BRASIL, B. C. DO. Cotação - Dólar EUA. In. Disponível em: <<https://www.bcb.gov.br/>>. Acesso em: 12 nov. 2021.
- CAMEO. Medium density fiberboard (MDF). In. Disponível em: <[http://cameo.mfa.org/wiki/Medium_density_fiberboard_\(MDF\)](http://cameo.mfa.org/wiki/Medium_density_fiberboard_(MDF))>. Acesso em: 12 nov. 2021.
- CASSIANO, C. et al. Desmotivação acadêmica: buscando compreender a realidade. *REFACS*, Uberaba, MG, v. 9, n. 2, p. 417–26, abr. 2021. Disponível em: <<http://seer.uftm.edu.br/revistaeletronica/index.php/refacs/article/view/4577>>. Acesso em: 27 out. 2021.
- CHENG, W. Leg Structure Design for Special Hexapod Robot with Climbing Vertical Walls Ability. In: 2020 International Conference on Electrical Engineering and Control Technologies (CEEET). [S.l.: s.n.], 2020. p. 1–4.

- DELGADO ESTEBAN, D. et al. Statically stable hexapod robot body construction. In: 2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE). [S.l.: s.n.], 2016. p. 1–4.
- EDUBOT. Protótipo de uma plataforma robótica livre para educação utilizando Metareciclagem. In. Disponível em: <<https://uriedubot.wordpress.com/>>. Acesso em: 23 out. 2021.
- ELETROGATE. Módulo Regulador de Tensão DC-DC LM317. In. Disponível em: <<https://www.eletrogate.com/modulo-regulador-de-tensao-dc-dc-lm317>>. Acesso em: 12 nov. 2021.
- ELETROGATE. Módulo Regulador De Tensão de 9V para 5V AMS1117. In. Disponível em: <<https://www.eletrogate.com/gap6n58wv-modulo-regulador-de-tensao-de-9v-para-5v-ams1117>>. Acesso em: 12 nov. 2021.
- ELETROGATE. Módulo Regulador de Tensão Step Down XL4005 - Conversor DC/DC 5A. In. Disponível em: <<https://www.eletrogate.com/modulo-regulador-de-tensao-step-down-xl4005-conversor-dc-dc-5a>>. Acesso em: 12 nov. 2021.
- ELETROGATE. Servo TowerPro MG946R. In. Disponível em: <<https://www.eletrogate.com/servo-towerpro-mg946r>>. Acesso em: 12 nov. 2021.
- ELETROGATE. Servo TowerPro MG995 Metálico 180°. In. Disponível em: <<https://www.eletrogate.com/servo-towerpro-mg995-metalico-180>>. Acesso em: 12 nov. 2021.
- ELETROGATE. Servo TowerPro MG996R. In. Disponível em: <<https://www.eletrogate.com/servo-towerpro-mg996r>>. Acesso em: 12 nov. 2021.
- ELETRÔNICA, B. DA. Micro-Servo. In. Disponível em: <<https://www.baudaeletronica.com.br/motores/micro-servo>>. Acesso em: 12 nov. 2021.
- ELETRÔNICA, B. DA. Placa Wemos D1 R32 ESP32. In. Disponível em: <<https://www.baudaeletronica.com.br/placa-wemos-d1-r32-esp32.html>>. Acesso em: 12 nov. 2021.
- FLIGHTMANIA. Ubec Hobbywing RC 3a 5v 6v Max 5a. In. Disponível em: <<https://www.flightmania.com.br/modulo-ubec-hobbywing-rc-3a-5v-6v-max-5a>>. Acesso em: 12 nov. 2021.
- FLOP, F. Módulo Wifi ESP8266 + Módulo de Expansão ESP-12F. In. Disponível em: <<https://www.filipeflop.com/produto/modulo-wifi-esp8266-modulo-de-expansao-esp-12f/>>. Acesso em: 12 nov. 2021.
- FLOP, F. Regulador de Tensão LM2596 Conversor DC-DC Step Down. In. Disponível em: <<https://www.filipeflop.com/produto/regulador-de-tensao-lm2596-conversor-dc-dc-step-down/>>. Acesso em: 12 nov. 2021.
- GUIMARÃES JÚNIOR, C. S. S. *Proposta De Um Framework Baseado Em Arquitetura Orientada a Serviços Para a Robótica*. 2015. f. 144. Dissertação (Mestrado em Engenharia Elétrica – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre).
- INDÚSTRIA, C. N. DA. Ensino de engenharia : fortalecimento e modernização /. *Confederação Nacional da Indústria.*, Brasília, v. 7, p. 32, 2018.


- KEBRITCHI, A.; HAVASHINEZHADIAN, S.; ROSTAMI, M. Design and Experimental Development of Hexapod Robot with Fiberglass-Fibercarbon Composite Legs. In: 2018 6th RSI International Conference on Robotics and Mechatronics (IcRoM). [S.l.: s.n.], 2018. p. 439–444.
- LEGO. Sobre LEGO MINDSTORMS. In. Disponível em: <<https://www.lego.com/pt-br/themes/mindstorms/about>>. Acesso em: 23 out. 2021.
- LIVRE, M. Esp32 Cam E Câmera Ov2640 De 2 Mega Pixels Wifi E Bluetooth. In. Disponível em: <https://produto.mercadolivre.com.br/MLB-1950979494-esp32-cam-e-cmera-ov2640-de-2-mega-pixels-wifi-e-bluetooth-_JM>. Acesso em: 12 nov. 2021.
- LIVRE, M. Esp32 Esp-32s Nodemcu Development Board 2.4gh Wifi Bluetooth. In. Disponível em: <https://produto.mercadolivre.com.br/MLB-1597140122-esp32-esp-32s-nodemcu-development-board-24gh-wifi-bluetooth-_JM>. Acesso em: 12 nov. 2021.
- LIVRE, M. Robô De Esteiras - Kit Robótico Educacional Para Arduino. In. Disponível em: <https://produto.mercadolivre.com.br/MLB-2081795168-rob-de-esteiras-kit-robotico-educacional-para-arduino-_JM>. Acesso em: 12 nov. 2021.
- MING, A. et al. Development of a novel leg mechanism in the shape of backward convex with a mechanical joint stop for bipedal robot. In: 2014 IEEE International Conference on Mechatronics and Automation. [S.l.: s.n.], 2014. p. 107–112.
- ÔMEGA, A. Kit Arduino Start Robô Controlado por App. In. Disponível em: <<https://www.arduinoomega.com/kit-start-robo-controlado-por-app>>. Acesso em: 12 nov. 2021.
- ÔMEGA, A. Micro Servo SG92R 9g TowerPro. In. Disponível em: <<https://www.arduinoomega.com/micro-servo-sg92r-9g-towerpro>>. Acesso em: 12 nov. 2021.
- ROBOCORE. Regulador de Tensão 5V - 7805. In. Disponível em: <<https://www.robocore.net/regulador-de-tensao/regulador-de-tensao-5v-7805>>. Acesso em: 12 nov. 2021.
- SILVA, A. F. DA. *RoboEduc: Uma Metodologia de Aprendizado com Robótica Educacional*. 2009. f. 115. Tese (Doutorado em Engenharia da Computação – Universidade Federal do Rio Grande do Norte, Lagoa Nova.
- STUDIO, S. ESP32-CAM Development Board(with camera). In. Disponível em: <<https://www.seeedstudio.com/ESP32-CAM-Development-Board-with-camer-p-3153.html>>. Acesso em: 12 nov. 2021.
- SUNFOUNDER. SunFounder Crawling Quadruped Robot Kit V2.0 for Arduino. In. Disponível em: <<https://www.sunfounder.com/products/arduino-crawling-quadruped-robot>>. Acesso em: 12 nov. 2021.
- SUPPLIES, P. Servo Motor Micro SG92R. In. Disponível em: <<https://protosupplies.com/product/servo-motor-micro-sg92r/>>. Acesso em: 12 nov. 2021.
- TANG, K. et al. Design and Implementation of Push Notification System Based on the MQTT Protocol. In: ISCA 2013. [S.l.: s.n.], 2013.

TEIXEIRA, E. A. S. *Arquitetura IOT para aplicação em automação residencial*. 2019. f. 51. Trabalho de conclusão de curso – Universidade Federal do Rio Grande do Sul, Porto Alegre.

YUAN, M. Conhecendo o MQTT. In. Disponível em: <<https://developer.ibm.com/br/articles/iot-mqtt-why-good-for-iot/>>. Acesso em: 21 out. 2021.

APÊNDICE A - FORMULÁRIO UTILIZADO NA PESQUISA COM DISCENTES


Pesquisa - Projeto de Extensão: Robótica Educacional

 Rascunho salvo.

***Obrigatório**

Qual seu ano de ingresso na faculdade? *

Sua resposta _____

 Esta pergunta é obrigatória

Você já se sentiu desmotivad@ e/ou com vontade de trocar de curso?

1 2 3 4 5

Nunca. Nosso curso é perfeito como é Sempre. Tô quase entrando com processo de transferência

Você gostou de interagir com o Edubot na cadeira de Introdução à Engenharia de Controle e Automação?

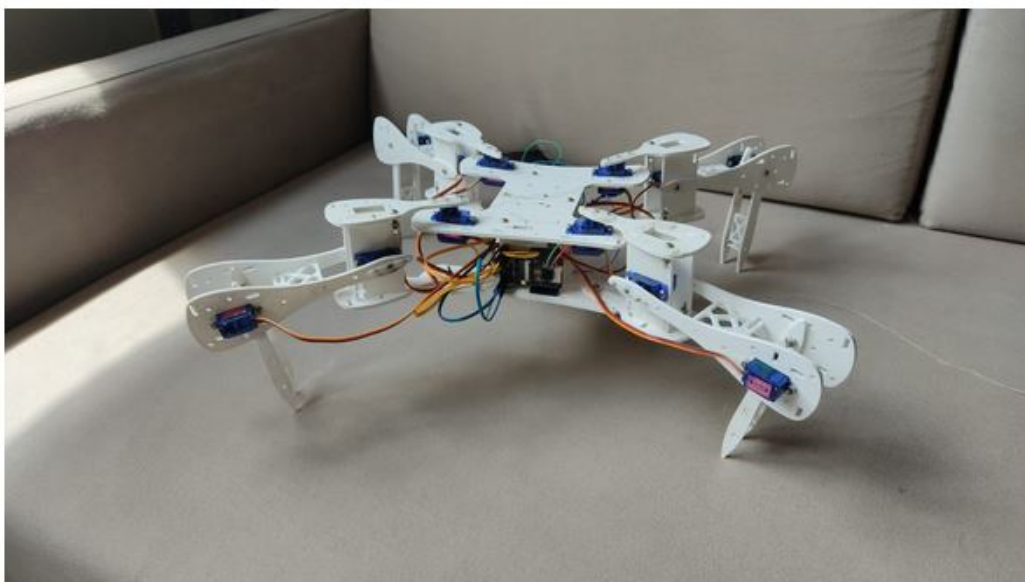
- Sim
- Não
- Mais ou menos
- Não cheguei a interagir

Você se sentiria mais motivad@ caso tivesse atividades de extensão com um foco mais em desenvolvimentos de projeto de robótica na prática, indo de modelagem em CAD, pra confecção, esquema elétrico/eletrônico até rotinas de programação?

- 1 2 3 4 5
- Não faria diferença na real. Iria me sentir muito mais motivad@!

Proposta de Atividade

O robô apresentado aqui na imagem é uma plataforma em que os servomotores podem ser montados em diferentes configurações, com membros modulares, possibilitando fazer mais uma "aranha" (como a configuração da imagem) ou um arranjo de membros mais de encontro a um "cachorro/mamífero", como o Spot, da Boston Dynamics. Tem uma câmera com o microprocessador e tem também WiFi e Bluetooth, permitindo fazer comunicação com aparelhos móveis por protocolos diferentes e também rodar rotinas de visão computacional.



Considerando a plataforma apresentada na imagem, dê uma nota de 1 a 5 pra sua motivação em escolher uma atividade de extensão que fosse desenvolver rotinas pra movimentação do robô de acordo com a configuração montada, assim como fazer novas "patas" e cortar em MDF ali na UFRGS pra testar diferentes tipos de movimentação.

- 1 2 3 4 5
-

Considerando a atividade proposta anteriormente, envolvendo desenho em CAD, protocolos de comunicação, cinemática inversa e programação de microcontroladores, o quanto você acha que ela agregaria positivamente à sua formação como profissional?

- 1 2 3 4 5
-

Por fim, o quanto você considera que essa atividade teria um impacto positivo na sua motivação para com o curso caso fosse realizada nos semestres iniciais do curso, entre 3 e 6?

- 1 2 3 4 5
-

APÊNDICE B - CÓDIGO IMPLEMENTADO EM C++

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

#define I2C_SDA 14
#define I2C_SCL 15
#define PIN_LED 4

#define SERVOMIN 150
#define SERVOMAX 600

#define ID_MQTT "Zambotto13"

/* Definindo os topicos */
#define TOPICO_SUBSCRIBE_LED "topico_liga_desliga_led"
#define TOPICO_SUBSCRIBE_PWM "topico_manda_pwn"
#define TOPICO_MOVE_OUTER_SERVOS "topico_move_outer_servos"
#define TOPICO_MOVE_MIDDLE_SERVOS "topico_move_middle_servos"
#define TOPICO_MOVE_INNER_SERVOS "topico_move_inner_servos"
#define TOPICO_STAND_UP "topico_stand_up"

/* Definicoes da estrutura fisica do robo*/
#define NUM_LIMBS 4
#define F_0 0
#define F_1 1
#define F_2 14
#define F_3 15
#define K_0 2
#define K_1 3
#define K_2 12
#define K_3 13
#define T_0 4
#define T_1 5
#define T_2 10
#define T_3 11
```

```

#define F_0_OFF 5
#define F_1_OFF 0
#define F_2_OFF 10
#define F_3_OFF 0
#define K_0_OFF -7
#define K_1_OFF 0
#define K_2_OFF -15
#define K_3_OFF 5
#define T_0_OFF -5
#define T_1_OFF 5
#define T_2_OFF -3
#define T_3_OFF -5

#define L2 4
#define L3 10
#define L4 10

/* Declaracao de estruturas*/
struct joint
{
    int servo_num;
    int servo_offset;
    bool bIsInverted;
};

struct limb
{
    joint joints[3];
};

struct body
{
    limb limbs[4];
};

struct limbPosition
{
    int thighAngle;
    int kneeAngle;
    int footAngle;
};

const char* SSID = "TellMyWifiMissHerToo";
const char* PASSWORD = "*****"; // Ocultado por segurança da
informação
TwoWire I2CA = TwoWire(0);
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40, I2CA);

const char* BROKER_MQTT = "mqtt.eclipseprojects.io";
int BROKER_PORT = 1883;

```

```
WiFiClient espClient;
PubSubClient MQTT(espClient);

joint foot0
{
  F_0,
  F_0_OFF,
  true
};
joint foot1
{
  F_1,
  F_1_OFF,
  false
};
joint foot2
{
  F_2,
  F_2_OFF,
  true
};
joint foot3
{
  F_3,
  F_3_OFF,
  false
};
joint knee0
{
  K_0,
  K_0_OFF,
  false
};
joint knee1
{
  K_1,
  K_1_OFF,
  true
};
joint knee2
{
  K_2,
  K_2_OFF,
  false
};
joint knee3
{
  K_3,
  K_3_OFF,
  true
};
```

```
joint thigh0
{
  T_0,
  T_0_OFF,
  false
};
joint thigh1
{
  T_1,
  T_1_OFF,
  true
};
joint thigh2
{
  T_2,
  T_2_OFF,
  true
};
joint thigh3
{
  T_3,
  T_3_OFF,
  false
};

joint all_servos[12] = {foot0, foot1,foot2, foot3, knee0, knee1, knee2,
  knee3, thigh0, thigh1, thigh2,thigh3};

static limb limb0
{
  {thigh0, knee0, foot0}
};
static limb limb1
{
  {thigh1, knee1, foot1}
};
static limb limb2
{
  {thigh2, knee2, foot2}
};
static limb limb3
{
  {thigh3, knee3, foot3}
};

static body robot0
{
  {limb0, limb1, limb2, limb3}
};

limbPosition currentAngles[4]
```

```

{
  {0,0,0},{0,0,0},{0,0,0},{0,0,0}
};

/* Prototypes */
void initWiFi(void);
void initMQTT(void);
void reconnectMQTT(void);
void reconnectWiFi(void);
void VerificaConexoesWiFIEMQTT(void);
void mqtt_callback(char* topic, byte* payload, unsigned int length);
void moveJointAngle(joint _joint, int angle);
void inverseKinematic(float x,float y, int* a1, int* a2);
void setLimbPosition(limb _limb, limbPosition desired_position);
void StepLimb(int SteppingLimb,int angleRaiseCompServos, int
  angleRaiseStepLeg, int angleForwardStep);

void initWiFi(void)
{
  delay(10);
  Serial.println("-----Conexao WI-FI-----");
  Serial.print("Conectando-se na rede: ");
  Serial.println(SSID);
  Serial.println("Aguarde");

  reconnectWiFi();
}

void initMQTT(void)
{
  MQTT.setServer(BROKER_MQTT, BROKER_PORT);
  MQTT.setCallback(mqtt_callback);
}

void mqtt_callback(char* topic, byte* payload, unsigned int length)
{
  String cmd;
  String value;

  /* obtem o comando */
  for(int i = 0; i < 5; i++)
  {
    char c = (char)payload[i];
    cmd += c;
  }

  /* obtem o valor do comando */

```

```

for(int i = 5; i < length; i++)
{
    char c = (char)payload[i];
    value += c;
}

Serial.print("Chegou o seguinte string via MQTT: ");
Serial.println(cmd + value);

/* toma aÃ§Ã£o dependendo da string recebida */
if (cmd.equals("led_"))
{
    if (value.equals("0"))
    {
        digitalWrite(PIN_LED, LOW);
        Serial.print("LED apagado mediante comando MQTT");
    }

    else if (value.equals("1"))
    {
        digitalWrite(PIN_LED, HIGH);
        Serial.print("LED aceso mediante comando MQTT");
    }
}

if (cmd.equals("mouts"))
{
    Serial.println("Mexendo servos via comando MQTT: 0, 1, 14, 15");
    Serial.println("Angulo: " + value);
    moveJointAngle(foot0, value.toInt());
    moveJointAngle(foot1, value.toInt());
    moveJointAngle(foot2, value.toInt());
    moveJointAngle(foot3, value.toInt());
    int i; for (i=0;i<4;i++)
    {
        currentAngles[i].footAngle = value.toInt();
    }
}

if (cmd.equals("mmids"))
{
    Serial.println("Mexendo servos via comando MQTT: 2, 3, 12, 13");
    Serial.println("Angulo: " + value);
    moveJointAngle(knee0, value.toInt());
    moveJointAngle(knee1, value.toInt());
    moveJointAngle(knee2, value.toInt());
    moveJointAngle(knee3, value.toInt());
    int i; for (i=0;i<4;i++)
    {
        currentAngles[i].kneeAngle = value.toInt();
    }
}

```

```

}

if (cmd.equals("minns"))
{
    Serial.println("Mexendo servos via comando MQTT: 4, 5, 10, 11");
    Serial.println("Angulo: " + value);
    moveJointAngle(thigh0, value.toInt());
    moveJointAngle(thigh1, value.toInt());
    moveJointAngle(thigh2, value.toInt());
    moveJointAngle(thigh3, value.toInt());
    int i; for (i=0;i<4;i++)
    {
        currentAngles[i].thighAngle = value.toInt();
    }
}

if (cmd.equals("setof"))
{
    char value_array[value.length()];
    value.toCharArray(value_array, value.length());
    char* servo = strtok(value_array, " _.'\0'");
    char* angle = strtok(NULL, " _.'\0'");
    Serial.print("Ajustando offset de servo ");
    Serial.print(servo);
    Serial.print(" em ");
    Serial.println(angle);
    int i;
    for (i=0;i<12;i++)
    {
        if (all_servos[i].servo_num == atoi(servo))
        {
            all_servos[i].servo_offset = atoi(angle);
        }
    }
}

if (cmd.equals("moves"))
{
    char value_array[value.length()];
    value.toCharArray(value_array, value.length());
    char* servo = strtok(value_array, " _.'\0'");
    char* angle = strtok(NULL, " _.'\0'");
    Serial.println("Movendo servo ");
    Serial.print(servo);
    Serial.print(" em ");
    Serial.println(angle);
    Serial.println(atoi(servo));
    Serial.println(atoi(angle));
    int i;
    for (i=0;i<12;i++)
    {

```



```

    if (all_servos[i].servo_num == atoi(servo))
    {
        Serial.println("entrou");

        moveJointAngle(all_servos[i], atoi(angle));

    }
}

if (cmd.equals("repor"))
{

    Serial.println("Fazendo report");

    int i;
    for (i=0;i<12;i++)
    {
        Serial.print(all_servos[i].servo_num);
        Serial.print(" -->");
        Serial.println(all_servos[i].servo_offset);

    }
}

if (cmd.equals("ikind"))
{

    Serial.println("Movendo limb individual com cinemática inversa");
    int angulomid, anguloout;
    char value_array[value.length()];
    value.toCharArray(value_array, value.length());
    char* limb1 = strtok(value_array, " _.'\0'");
    char* x1 = strtok(NULL, " _.'\0'");Serial.print(x1);
    char* y1 = strtok(NULL, " _.'\0'");Serial.print(y1);
    int limb = atoi(limb1);
    int x = atoi(x1);
    int y = atoi(y1);
    Serial.print("X =");
    Serial.print(x);
    Serial.print(" Y =");
    Serial.println(y);

    inverseKinematic(x,y,&angulomid, &anguloout);
    Serial.print("anguloMid =");
    Serial.print(angulomid);
    Serial.print(" anguloout =");
    Serial.println(anguloout);

    limbPosition position;
    position.thighAngle = 90;

```

```

    position.kneeAngle = angulomid;
    position.footAngle = anguloout;

    setLimbPosition(robot0.limbs[limb],position);

}

if (cmd.equals("stand"))
{
    if (value.equals("0"))
    {
        Serial.println("Standing down");
        delay(500);
        int angle = 70;
        moveJointAngle(knee0, angle);
        moveJointAngle(knee1, angle);
        moveJointAngle(knee2, angle);
        moveJointAngle(knee3, angle);
        int i; for (i=0;i<4;i++)
        {
            currentAngles[i].kneeAngle = angle;
        }
        delay(500);

        angle = 60;
        moveJointAngle(knee0, angle);
        moveJointAngle(knee1, angle);
        moveJointAngle(knee2, angle);
        moveJointAngle(knee3, angle);
        for (i=0;i<4;i++)
        {
            currentAngles[i].kneeAngle = angle;
        }
    }

    else if (value.equals("1"))
    {
        Serial.println("Standing up");

        int angle = 90;
        moveJointAngle(foot0, angle);
        moveJointAngle(foot1, angle);
        moveJointAngle(foot2, angle);
        moveJointAngle(foot3, angle);
        int i; for (i=0;i<4;i++)
        {
            currentAngles[i].footAngle = angle;
        }
        delay(300);

        angle = 100;
    }
}

```

```

    moveJointAngle(knee0, angle);
    moveJointAngle(knee1, angle);
    moveJointAngle(knee2, angle);
    moveJointAngle(knee3, angle);
    for (i=0;i<4;i++)
    {
        currentAngles[i].kneeAngle = angle;
    }
}

else if (value.equals("2"))
{
    int angle = 90;
    moveJointAngle(foot0, angle);
    moveJointAngle(foot1, angle);
    moveJointAngle(foot2, angle);
    moveJointAngle(foot3, angle);
    int i; for (i=0;i<4;i++)
    {
        currentAngles[i].footAngle = angle;
    }
    delay(300);

    moveJointAngle(thigh0, 120);
    currentAngles[0].thighAngle = 120;
    moveJointAngle(thigh1, 120);
    currentAngles[1].thighAngle = 120;
    moveJointAngle(thigh2, 60);
    currentAngles[2].thighAngle = 60;
    moveJointAngle(thigh3, 60);
    currentAngles[3].thighAngle = 60;
    delay(500);

    angle = 100;
    moveJointAngle(knee0, angle);
    moveJointAngle(knee1, angle);
    moveJointAngle(knee2, angle);
    moveJointAngle(knee3, angle);
    for (i=0;i<4;i++)
    {
        currentAngles[i].kneeAngle = angle;
    }
}
}

if (cmd.equals("step1"))
{
    int i;
    int limb = value.toInt();
    StepLimb(limb,5,30,40);
}

```

```

}

if (cmd.equals("walk_"))
{
    int n_reps = value.toInt();
    int i = 0;
    int raiseComp = 10;
    int raiseStep = 30;
    int anglefw = 40;

    while (i< n_reps)
    {
        StepLimb(1,raiseComp,raiseStep,anglefw);
        delay(200);
        StepLimb(3,raiseComp,raiseStep,anglefw);
        delay(200);
        StepLimb(0,raiseComp,raiseStep,anglefw);
        delay(200);
        StepLimb(2,raiseComp,raiseStep,anglefw);
        delay(200);
        i++;
    }
}

}

void StepLimb(int SteppingLimb,int angleRaiseCompServos, int
    angleRaiseStepLeg, int angleForwardStep)
{
    int i;
    for(i=0;i<NUM_LIMBS;i++)
    {
        if (i != SteppingLimb)
        {
            moveJointAngle(robot0.limbs[i].joints[1],currentAngles[i].
kneeAngle-angleRaiseCompServos);
            currentAngles[i].kneeAngle += -angleRaiseCompServos;
        }
    }
    delay(500);

    for(i=0;i<NUM_LIMBS;i++)
    {
        if (i == SteppingLimb)
        {
            moveJointAngle(robot0.limbs[i].joints[1],currentAngles[i].
kneeAngle-angleRaiseStepLeg);
            delay(300);

```

```

        moveJointAngle(robot0.limbs[i].joints[0],currentAngles[i].
thighAngle +angleForwardStep);
        delay(300);
        moveJointAngle(robot0.limbs[i].joints[1],currentAngles[i].
kneeAngle);
        delay(300);
        currentAngles[i].thighAngle += angleForwardStep;
    }
}

for(i=0;i<NUM_LIMBS;i++)
{
    if (i != SteppingLimb)
    {
        moveJointAngle(robot0.limbs[i].joints[1],currentAngles[i].
kneeAngle+angleRaiseCompServos);
        currentAngles[i].kneeAngle += angleRaiseCompServos;
        delay(50);
    }
}
delay(300);

for(i=0;i<NUM_LIMBS;i++)
{
    moveJointAngle(robot0.limbs[i].joints[0],currentAngles[i].
thighAngle -(angleForwardStep/4));
    currentAngles[i].thighAngle -=(angleForwardStep/4);
}
}

void inverseKinematic(float x,float y, int* a1, int* a2)
{
    double U = sqrt((x*x)+(y*y));
    double gamma = atan2(x,y);
    double gamma_com_alfa;
    if (((B*B)+(U*U)-(C*C))/(2*B*U) < 1)
        gamma_com_alfa = acos(((B*B)+(U*U)-(C*C))/(2*B*U));
    else
        gamma_com_alfa = acos(1);
    double alfa = gamma_com_alfa - gamma;
    double beta;
    if ((U*sin(gamma_com_alfa)/C) < 1)
        beta = asin(U*sin(gamma_com_alfa)/C);
    else
        beta = asin(1);
    if (U > sqrt((B*B)+(C*C)))
    {
        beta =(PI) - beta;
    }
}

```

```

int anglemid = 90- int(alfa * (180/PI));
int angleout = int(beta * (180/PI));
if ((anglemid>0) && (anglemid<180)){
    *a1=anglemid;
}
if ((angleout>0) && (angleout<180)){
    *a2=angleout;
}
}

void moveJointAngle(joint _joint, int degree)
{
    int internal_degree = degree + _joint.servo_offset;
    if (_joint.bIsInverted)
    {
        internal_degree = 180 - internal_degree;
    }
    pwm.setPWM(_joint.servo_num,0,(map(internal_degree,0,180,SERVOMIN,
SERVOMAX)));
}

void setLimbPosition(limb _limb, limbPosition desired_position) {
    moveJointAngle(_limb.joints[0], desired_position.thighAngle);
    moveJointAngle(_limb.joints[1], desired_position.kneeAngle);
    moveJointAngle(_limb.joints[2], desired_position.footAngle);
}

void reconnectMQTT()
{
    while (!MQTT.connected())
    {
        Serial.print("* Tentando se conectar ao Broker MQTT: ");
        Serial.println(BROKER_MQTT);
        if (MQTT.connect(ID_MQTT))
        {
            Serial.println("Conectado com sucesso ao broker MQTT!");
            MQTT.subscribe(TOPICO_SUBSCRIBE_LED);
            MQTT.subscribe(TOPICO_SUBSCRIBE_PWM);
            MQTT.subscribe(TOPICO_MOVE_OUTER_SERVOS);
            MQTT.subscribe(TOPICO_MOVE_MIDDLE_SERVOS);
            MQTT.subscribe(TOPICO_MOVE_INNER_SERVOS);
            MQTT.subscribe(TOPICO_STAND_UP);
        }
        else
        {
            Serial.println("Falha ao reconectar no broker.");
            Serial.println("Havera nova tentatica de conexao em 2s");
            delay(2000);
        }
    }
}

```

```

    }
}

void VerificaConexoesWiFIEMQTT(void)
{
    if (!MQTT.connected())
        reconnectMQTT();

    reconnectWiFi();
}

void reconnectWiFi(void)
{
    if (WiFi.status() == WL_CONNECTED)
        return;

    WiFi.begin(SSID, PASSWORD);

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }

    Serial.println();
    Serial.print("Conectado com sucesso na rede ");
    Serial.print(SSID);
    Serial.println("IP obtido: ");
    Serial.println(WiFi.localIP());
}

/* Função de setup */
void setup()
{
    Serial.begin(115200);
    Wire.begin(I2C_SDA, I2C_SCL);
    pinMode(PIN_LED, OUTPUT);
    digitalWrite(PIN_LED, LOW);
    I2CA.begin(I2C_SDA, I2C_SCL, 100000);

    pwm.begin();
    pwm.setOscillatorFrequency(27000000);
    pwm.setPWMFreq(50);
    Wire.setClock(400000);

    initWiFi();
}

```

```
    initMQTT();  
}  
  
/* Loop principal */  
void loop()  
{  
    VerificaConexoesWiFiMQTT();  
    MQTT.loop();  
}
```
