

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ENGENHARIA
CAMPUS DE ILHA SOLTEIRA

CAMILA MOMESSO

PROJETO DE HARDWARE PARA ATUALIZAÇÃO DO CONTROLADOR DE UM
MANIPULADOR SCORBOT UTILIZANDO O PROTOCOLO CAN

Ilha Solteira, SP

2023

CAMILA MOMESSO

PROJETO DE HARDWARE PARA ATUALIZAÇÃO DO CONTROLADOR DE UM
MANIPULADOR SCORBOT UTILIZANDO O PROTOCOLO CAN

Trabalho Conclusão do Curso de Graduação
apresentado à Faculdade De Engenharia de Ilha
Solteira – UNESP como parte dos requisitos
para obtenção do título de engenheiro
mecânico.

Orientador: Prof. Dr. Amarildo Tabone
Paschoalini

Ilha Solteira

2023

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

Momesso, Camila.

M732p Projeto de hardware para a atualização do controlador de um manipulador SCORBOT utilizando o protocolo CAN / Camila Momesso. -- Ilha Solteira: [s.n.], 2023

47 f. : il.

Trabalho de conclusão de curso (Graduação em Engenharia Mecânica) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2023

Orientador: Amarildo Tabone Paschoalini

Inclui bibliografia

1. Braço robótico. 2. Protocolo CAN. 3. Microcontroladores. 4. Ensino de robótica. 5. Controlador aberto. 6. Arquitetura descentralizada.

Raimundo da Silva Santos
Raimundo da Silva Santos

Supervisor Técnico de Biblioteca e Documentação
Bibliotecária de Referência, Atendimento ao usuário e Documentação
Classe: 1999

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

FACULDADE DE ENGENHARIA - CAMPUS DE ILHA SOLTEIRA

CURSO DE ENGENHARIA MECÂNICA

ATA DA DEFESA – TRABALHO DE GRADUAÇÃO

TÍTULO: Projeto de hardware para a atualização do controlador de um manipulador SCORBOT utilizando o protocolo CAN.

ALUNA: Camila Momesso RA:162054238

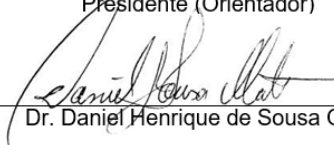
ORIENTADOR: Prof. Dr. Amarildo Tabone Paschoalini

Aprovado (X) - Reprovado () pela Comissão Examinadora

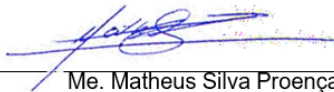
Comissão Examinadora:



Prof. Dr. Amarildo Tabone Paschoalini
Presidente (Orientador)



Dr. Daniel Henrique de Sousa Obata



Me. Matheus Silva Proença

Ilha Solteira (SP) 10 de julho de 2023.

AGRADECIMENTOS

Agradeço primeiro aos meus pais Eder Momesso e Claudenice Aparecida Collangelli Momesso que sempre estiveram ao meu lado me apoiando ao longo de toda a minha trajetória.

Ao meu orientador Prof. Dr. Amarildo Tabone Paschoalini por aceitar conduzir o meu trabalho de pesquisa e por fornecer todo o auxílio necessário ao longo do desenvolvimento desse trabalho.

Agradeço às minhas amigas, Letícia Urani, Elaine Garcia e Lorena Vaz e meu irmão Eder Momesso que sempre me apoiaram e incentivaram desde o início de minha pesquisa. Também ao Prof. Dr. Daniel Obata por compartilhar comigo sua experiência e conhecimento durante o projeto.

À UNESP de Ilha Solteira e todos os professores e funcionários que contribuíram de forma direta ou indireta para a realização dessa conquista.

“Qualquer tecnologia suficientemente avançada é indistinguível da magia”

Arthur C. Clarke

RESUMO

Este trabalho tem como objetivo realizar uma atualização de software em um controlador de um manipulador robótico SCORBOT-ER VPlus utilizando uma arquitetura aberta e descentralizada para fins didáticos. A idealização do projeto surgiu como uma forma de aproveitar o Scorbót que se encontra em desuso na faculdade de engenharia de Ilha Solteira devido a desatualização do controlador desse robô, que, por apresentar um sistema operacional proprietário, oferece diversas limitações para de comunicar com tecnologias atuais.

Visando a elaboração de um mecanismo versátil e moderno para aplicações didáticas, foi proposta a utilização do protocolo CAN para estabelecer a comunicação entre DSPs responsáveis pelo controle de cada grau de liberdade do robô por meio dos microcontroladores STM32 e ESP32, para envio e recebimento das mensagens do barramento. A utilização dessas tecnologias foi definida com intuito de fornecer ferramentas modernas e importantes para capacitação dos alunos de graduação no contexto de avanços tecnológicos impulsionados por tecnologias emergentes.

Palavras-chave: Braço robótico. Protocolo CAN. Microcontroladores. Ensino de robótica. Controlador aberto. Arquitetura descentralizada. ESP32. STM32. Scorbót. Manipuladores robóticos.

ABSTRACT

This work aims to perform a software update on the controller of a SCORBOT-ER VPlus robotic manipulator using an open and decentralized architecture for educational purposes. The project idea arose as a way to utilize the Scrobot, which is no longer in use at the Engineering College of Ilha Solteira, due to the outdated controller of this robot. The proprietary operating system of the controller presents several limitations in communicating with current technologies.

With the objective of developing a versatile and modern mechanism for educational applications, the use of the CAN protocol was proposed to establish communication between DSPs responsible for controlling each degree of freedom of the robot using STM32 and ESP32 microcontrollers for message transmission and reception on the bus. The adoption of these technologies was defined with the aim of providing modern and important tools for the training of undergraduate students in the context of technological advancements driven by emerging technologies.

Keywords: Robotic arm. CAN protocol. Microcontrollers. Robotics education. Open controller. Decentralized architecture. ESP32. STM32. Scrobot. Robotic manipulators.

LISTA DE FIGURAS

Figura 1 – Bits dominantes e recessivos.....	19
Figura 2 – Estrutura de identificação do CAN de 11 bit.....	20
Figura 3 – Detalhes do barramento CAN.....	21
Figura 4 – Scrobot-ER Vplus.....	22
Figura 5 – Elos e juntas rotativas do Scobot.....	23
Figura 6 – Transmissão.....	24
Figura 7 – Área de trabalho do Scrobot vista lateral.....	25
Figura 8 – Área de trabalho pela vista superior.....	26
Figura 9 – Encoder.....	27
Figura 10 – Sinais captados pelo Encoder.....	28
Figura 11 – Microswitch.....	29
Figura 12 – Conector DB-50.....	30
Figura 13 – Esquema de comunicação entre as DSPs.....	32
Figura 14 – Diagrama de pinos STM32.....	34
Figura 15 – Pinos do driver L298N.....	36
Figura 16 – Driver L298N.....	37
Figura 17 – Módulo CAN BUS TJA1050.....	38
Figura 18 – Regulador de tensão Mini 360.....	38
Figura 19 – Conectores.....	39
Figura 20 – Esquemático das conexões dos módulos de controle dos atuadores.....	41
Figura 21 – Layout do circuito.....	42
Figura 22 – Trajetória da CNC especificada no FlatCAM.....	43
Figura 23 – Confeção dos módulos de controle dos atuadores para testes.....	44
Figura 24 – Solda realizada na placa de testes.....	45

LISTA DE QUADROS

Quadro 1 – Descrição do movimento das juntas do Scrobot.....	23
Quadro 2 – Especificações técnicas do Scrobot.....	26
Quadro 3– Cabeamento dos motores.....	30
Quadro 4– Cabeamento dos Encoders.....	31
Quadro 5– Cabeamento dos Microswitches.....	32
Quadro 6 – Configurações do diagrama de pinos do STM32.....	34
Quadro 7 – Funções dos pinos do STM32.....	41

LISTA DE ABREVIATURAS E SIGLAS

IoT	Internet of Things
CAN	Controller Area Network
CSMA	Carrier Sense Multiple Access
CD	Collision Detection
AMP	Arbitration on Message Priority
SOF	Start of frame
RTR	Remote Transmission Request
DLC	Data Length Code
CRC	Cyclic Redundancy Check
ACK	Acknowledges
EOF	end of frame
IFS	interframe space
DSP	Digital Signal Processor
PWM	Pulse Width Modulation

SUMÁRIO

1	INTRODUÇÃO	15
1.1	CONTEXTUALIZAÇÃO DO PROJETO.....	15
1.1.1	Objetivo principal.....	15
1.1.2	Justificativa.....	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	PROTOCOLO CAN	16
2.2	SCORBOT	21
2.1.1	Especificações técnicas	21
2.1.2	Encoders Ópticos.....	25
2.1.3	Chaves de fim de curso	27
2.1.4	Cabeamento	28
3	DESENVOLVIMENTO DA ELETRÔNICA DE CONTROLE.....	31
3.1	IDEALIZAÇÃO DO CIRCUITO.....	31
3.2	MICROCONTROLADOR	32
3.3	ESPECIFICAÇÃO DOS COMPONENTES	34
3.3.1	Ponte H.....	34
3.3.2	Transceptor TJA1050.....	36
3.3.3	Conversor DC-DC Step Down	37
3.3.4	Conectores.....	38
3.4	MÓDULO DE CONTROLE PRINCIPAL	38
4	RESULTADOS E DISCUSSÕES	39
4.1	CONFECÇÃO DOS MÓDULOS DE CONTROLE DOS ATUADORES	39
4.2	TRABALHOS FUTUROS.....	44
5	CONCLUSÃO	46
6	REFERÊNCIAS BIBLIOGRÁFICAS	47

1 INTRODUÇÃO

Buscando maior eficiência, produtividade e segurança, surgiu a necessidade de desenvolver, dentro das fábricas, uma tecnologia que automatizasse processos repetitivos e cansativos aos seres humanos. A solução para esses desafios foi encontrada na robótica, com destaque para os manipuladores industriais, devido à agilidade, precisão e amplo espaço de trabalho que esses dispositivos oferecem (SAHA, 2008).

Com a rápida evolução da Indústria 4.0, impulsionada pelas tecnologias emergentes, como Internet das Coisas (IoT), *Big Data* e Inteligência Artificial (IA), as aplicações robóticas estão passando por avanços significativos. Nesse contexto, a integração de sistemas robóticos com sensores avançados tem permitido aprimorar a eficiência e funcionalidade desses mecanismos (SCHLEGEL, 2017).

Para acompanhar esses avanços, os sistemas eletrônicos apresentam uma necessidade cada vez maior de conseguir trocar informações entre várias unidades de controle. Foi nesse contexto que, na década de 1980, a Bosch GmbH desenvolveu a tecnologia CAN (*Controller Area Network*), com o objetivo de proporcionar uma comunicação econômica e eficiente para sistemas automotivos, reduzindo significativamente a quantidade de cabos necessários para a troca de informação entre os diversos equipamentos de controle. Atualmente, essa tecnologia é amplamente utilizada em diversas outras aplicações, como dispositivos médicos, sistemas de controle industrial e sistemas aeroespaciais (Di Natale, 2017).

1.1 CONTEXTUALIZAÇÃO DO PROJETO

1.1.1 Objetivo principal

Acompanhando as evoluções tecnológicas que buscam a automatização das linhas de produção, existe uma demanda cada vez mais alta de profissionais capacitados na área de robótica. Assim, esse projeto apresenta como objetivo principal desenvolver um sistema de controle utilizando código aberto para o braço mecânico Scorbot, visando fornecer uma ferramenta didática que possibilite o aprofundamento em aplicações práticas de robótica.

1.1.2 Justificativa

O Scorbob é um exemplo de manipulador amplamente utilizado como ferramenta didática no ensino da robótica. Contudo, assim como a maioria dos robôs industriais, o controlador do Scorbob apresenta um sistema operacional proprietário, o que limita o desenvolvimento da programação por meio de tecnologias atuais.

Ao adotar um sistema com arquitetura aberta, os estudantes têm a oportunidade de acessar o código fonte do projeto e desenvolver funcionalidades personalizadas para o mecanismo. Isso permite a consolidação de conhecimentos em programação, mecânica e eletrônica, proporcionando uma experiência de aprendizado mais abrangente e prática.

Além disso, a arquitetura aberta possibilita a incorporação de recursos mais atualizados ao sistema. Como o controlador original do Scorbob utiliza uma tecnologia desatualizada, existe uma limitação para realizar a integração com dispositivos modernos. Dessa forma o manipulador pode ser integrado a outras tecnologias, como Internet das Coisas (IoT), o que proporciona projetos de automação mais complexos.

Outra vantagem importante que justifica o desenvolvimento deste projeto seria o estímulo a inovação colaborativa. Com uma arquitetura aberta, a comunidade de desenvolvedores e usuários pode contribuir com melhorias e novos recursos ao sistema.

2 REVISÃO BIBLIOGRÁFICA

Nesta seção, serão apresentadas, de forma detalhada, as abordagens das tecnologias empregadas no desenvolvimento desse projeto. Objetivou-se, com a reunião dessas informações, proporcionar uma compreensão mais aprofundada das escolhas tecnológicas feitas durante a elaboração deste trabalho, permitindo uma visão abrangente e embasada das decisões tomadas ao longo do processo de desenvolvimento.

2.1 PROTOCOLO CAN

O aumento das expectativas em relação ao potencial de desenvolvimento de um sistema eletrônico multifuncional impulsionou uma transição de sistemas individuais separados com operação autônoma para um sistema totalmente integrado que resulta na interconexão de várias unidades de controle eletrônico. Um exemplo disso é o sistema de controle de tração dos

veículos, que desempenha uma função implementada pela ação conjunta das unidades de controle do motor e do sistema de freios ABS (SCHÄUFFELE, 2005).

Contudo, a implementação de várias tarefas quase simultâneas em um único processador pode apresentar uma variedade de desafios, tais como: conflito de recursos, limitações em desempenho e contenção de memória, gerenciamento de tempo, entre outros. Assim, a implementação de redes de comunicações entre múltiplos dispositivos conectados foi uma alternativa para contornar esses problemas. O fluxo de informações destes pode ocorrer de diferentes maneiras, sendo uma abordagem bem comum por meio de um barramento. Existem dois métodos de acesso ao barramento: centralizado e descentralizado (SCHÄUFFELE, 2005).

Na arquitetura centralizada, também denominada por “arquitetura master/slave” (mestre/escravo), existe um nó central da rede (mestre), responsável por controlar todas as unidades eletrônicas. Já a arquitetura descentralizada descreve uma implementação “multimestre” (Di Natale, 2017).

A escolha entre uma estratégia centralizada ou descentralizada depende do objetivo do projeto, sendo que ambas apresentam vantagens e desvantagens. No desenvolvimento deste trabalho, optou-se em utilizar uma arquitetura descentralizada, pois em comparação com sistemas centralizados, esses sistemas proporcionam vários benefícios, tais como: redução significativa nos custos de cabeamento, expansibilidade (devido a facilidade em realizar uma implementação modular de funções opcionais ao sistema) e maior confiabilidade e segurança dos sistemas.

Contudo, em uma arquitetura descentralizada, é preciso elaborar estratégias mais complexas de serem implementadas para contornar as colisões de mensagens no barramento. Neste trabalho foi adotado o protocolo CAN para desempenhar essa função.

O protocolo CAN é um protocolo de comunicação utilizando redes de controle e monitoramento de sistemas que permite uma comunicação confiável e eficiente entre os dispositivos conectados em um barramento CAN. Nesse barramento em comum, os dispositivos enviam e recebem mensagens identificadas por um identificador único e as colisões são evitadas com base nas propriedades atribuídas as mensagens, utilizando a abordagem CSMA/CD+AMP, sendo a denominação de cada termo da sigla apresentado a seguir (SCHÄUFFELE, 2005):

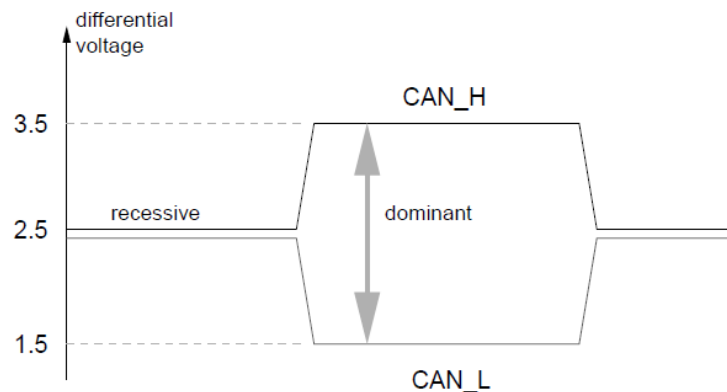
- CSMA (Carrier Sense Multiple Access): esse parâmetro define que cada nó em um barramento deve aguardar um período prescrito de inatividade antes de enviar uma mensagem.

- CD (Collision Detection) + AMP (Arbitration on Message Priority): confere a detecção de colisões que são solucionadas por meio de uma arbitragem bit a bit, baseando-se em uma prioridade pré-programada para cada mensagem no campo de identificação desta. Assim, a identificação com prioridade mais alta, ou seja, com o último bit lógico alto no identificador, sempre tem acesso ao barramento.

A rede CAN trabalha com dois fios que utilizam os sinais de dados CAN high e CAN low, assim, é possível identificar o estado lógico do sinal por meio de um diferencial da voltagem entre esses parâmetros. De acordo com o padrão CAN, os estados obtidos são denominados recessivo e dominante (NAVET, 2008).

Diferentemente das expectativas convencionais, o estado lógico do sinal recessivo é 1, e do dominante, 0 (NAVET, 2008). No estado recessivo, as duas linhas de sinal do barramento são polarizadas com, aproximadamente, 2.5 V. Já o estado dominante no barramento, o CANH aumenta para 3.5V e o CANL diminui para 1.5, como pode ser aferido na Figura 1.

Figura 1 – Bits dominantes e recessivos



Fonte: NAVET, 2008.

O acesso ao barramento é baseado em eventos que ocorrem de maneira aleatória. Como foi mencionado anteriormente, quando dois ou mais nós tentam enviar uma mensagem simultaneamente ao barramento, o acesso é realizado por meio de uma arbitragem não destrutiva. A característica não destrutiva significa que o nó que apresentar maior prioridade obtém acesso ao barramento para transmitir sua mensagem, já os demais nós aguardam sem que tenham suas mensagens corrompidas (SCHÄUFFELE, 2005).

Quanto menor o número binário do identificador, maior sua prioridade. Pode ser citado como exemplo essa alocação de prioridade o acionamento de um motor que apresenta o identificador 0010 para um sinal de feedback de corrente de enrolamento e 0011 a identificação do tacômetro. Neste cenário, como o identificador mais baixo é 0010, as mensagens relacionadas aos valores de corrente apresentam prioridade mais alta no barramento que as relacionadas a leitura do tacômetro.

Foi estabelecido pelo padrão ISO-11898:2003 (TEXAS INSTRUMENTS, 2016) o identificador padrão de 11 bits, ou seja, permite 2048 identificadores de mensagens diferentes e taxas de sinalização padrão de 125kbps a 1 Mbps. Posteriormente, o identificador foi estendido para 29 bits. A estrutura dos campos de identificação no quadro de mensagem do CAN pode ser aferida pela Figura 2.

Figura 2 – Estrutura de identificação do CAN de 11 bit

S O F	11-bit Identifier	R T R	I D E	r0	DLC	0...8 Bytes Data	CRC	ACK	E O F	I F S
----------------------	------------------------------	----------------------	----------------------	-----------	------------	-------------------------	------------	------------	----------------------	----------------------

Fonte: (TEXAS INSTRUMENTS, 2016).

O significado de cada campo de bits na Figura 2 pode ser descrito como:

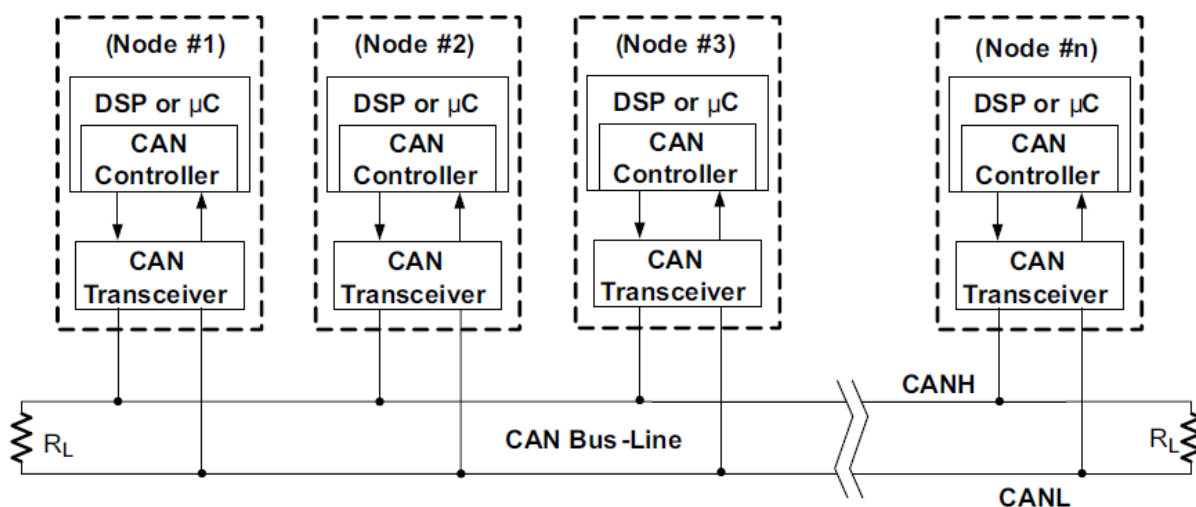
- SOF (*Start of frame*): identifica o início de uma mensagem e é utilizado para sincronizar os nós em um barramento após um período de inatividade.
- Identificador: estabelece a prioridade da mensagem.
- RTR (*remote transmission request*): bit único de solicitação de transmissão remota, indica se será uma mensagem de dado (*data message*) ou de resposta (*reply message*).
- IDE (*identifier extension*): significa que será transmitido um identificador padrão CAN, sem extensão.
- r0: bit reservado para possível uso em emendas futuras.
- DLC (*data length code*): o código de comprimento de dados de 4 bits contém o número de bytes de dados que estão sendo transmitidos.
- Dados: Até 64 bits de dados de aplicação podem ser transmitidos.
- CRC (*cyclic redundancy check*): esse ciclo de verificação de 16 bits é responsável

pela detecção de erros.

- ACK (*acknowledges*): formado por dois bits: de reconhecimento e delimitador. O de reconhecimento pode assumir valor de dominante e recessivo dependendo da mensagem transmitida. Já o valor do bit delimitador sempre será recessivo para indicar o fim desse campo.
- EOF (*end of frame*): composto por sete bits recessivos, indica o fim da transmissão de uma mensagem.
- IFS (*interframe space*): esse espaço de 7 bits contém o tempo necessário para finalizar o processamento interno do controlador CAN.

Já na Figura 3, é possível observar a arquitetura física de um sistema utilizando o protocolo CAN.

Figura 3 – Detalhes do barramento CAN



Fonte: TEXAS INSTRUMENTS, 2016.

A nomenclatura DSP (*Digital Signal Processor*) utilizada na Figura 3 refere-se a um microprocessador de sinal digital. É possível observar que o DSP utilizado apresenta um controlador CAN integrado de 3.3V. Já a comunicação com o barramento é implementada por meio de um transceptor de linha, formando um nó no sistema.

Como o sinal de comunicação no barramento CAN é diferencial, o sistema apresenta alta imunidade a ruídos e tolerância a falhas, pois o balanceamento faz com que a corrente que

flui em cada linha de sinal tenha a mesma intensidade, mas com direção oposta, resultando em um cancelamento de campo que filtra as baixas emissões de ruído. Essa característica também permite altas taxas de transmissão sobre pares trançados. De acordo com as especificações do padrão ISO 11898 High-Speed (TEXAS INSTRUMENTS, 2016) é estabelecido uma taxa de 1 Mbps com comprimento máximo de 40 metros e um máximo de 30 nós.

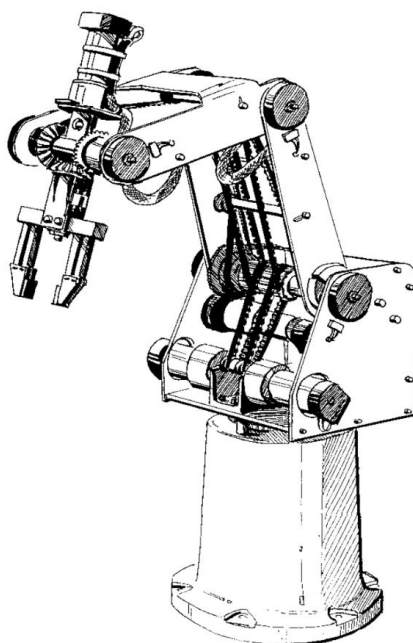
A impedância característica do cabo utilizado no barramento CAN costuma ser de 120 Ω (ohms). Assim, para evitar reflexão de sinal, são colocados resistores de 120 Ω nos dois extremos do cabo, denominados resistores de terminação, ilustrados na Figura 3 como RL.

2.2 SCORBOT

2.1.1 Especificações técnicas

O Scorbote-ER Vplus, assim como os demais manipuladores industriais, é antropomórfico, ou seja, semelhante ao braço humano, sendo possível identificar o tronco, braço, antebraço, ombro, cotovelo e punho desse mecanismo. Sua estrutura apresenta seis graus de liberdade e é composta por juntas rotativas e um efetuator do tipo garra com movimento prismático. Na Figura 4, é possível observar uma ilustração do manipulador.

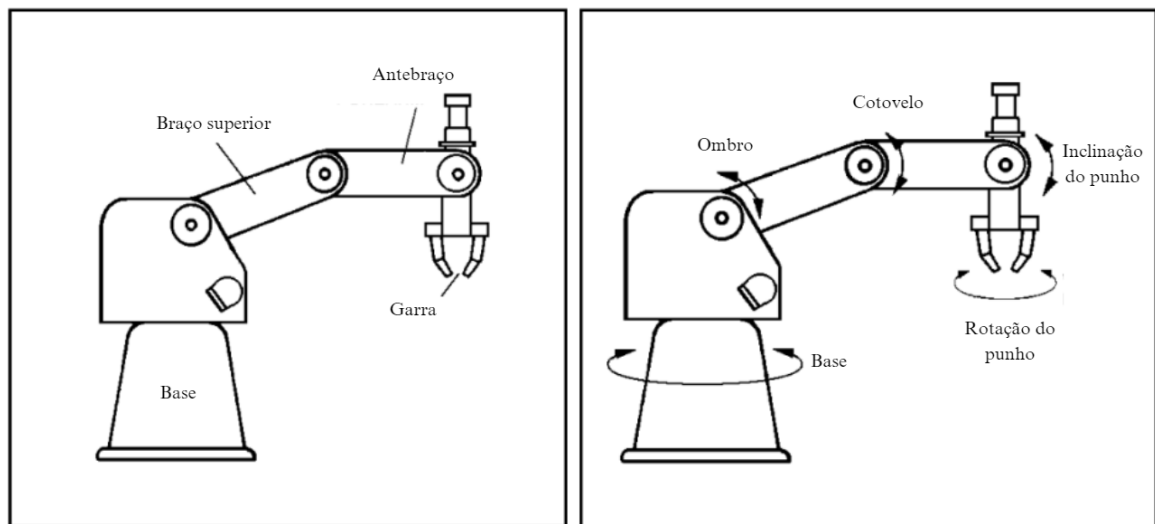
Figura 4 – Scorbote-ER Vplus



Fonte: [7].

Fabricado pela empresa Intelitek, o Scrobot-ER Vplus é um braço robótico industrial que conta com cinco motores DC de 12V acoplados a cada junta rotativa do manipulador, juntamente com um encoder óptico. Na garra, o Scrobot apresenta um sexto motor com as mesmas características, responsável pela sua abertura e fechamento. Pode ser aferido pela Figura 5 um esquemático desse mecanismo, apresentando os elos e movimentação das juntas do manipulador.

Figura 5 – Elos e juntas rotativas do Scrobot



Fonte: [7].

Para melhor entendimento das funcionalidades do Scrobot, os motores responsáveis pelos movimentos descritos na Figura 5 podem ser aferidos pela Tabela 2.1.

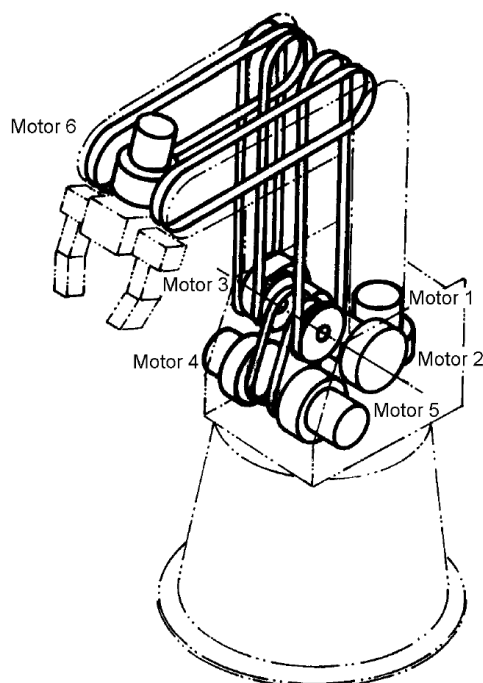
Quadro 1 – Descrição do movimento das juntas do Scrobot

Nome da junta	Descrição do movimento	Número do motor
Base	Rotação do corpo	1
Ombro	Subida e descida do braço	2
Cotovelo	Subida e descida do antebraço	3
Punho (Inclinação)	Subida e descida da garra	4+5
Punho (Rotação)	Rotação da garra	4+5

Fonte: Elaborado pela autora.

O posicionamento dos seis motores e a transmissão de movimento dos elos do braço robótico descritas na Quadro 1 podem ser aferidas pela Figura 6, sendo o movimento da base e do ombro realizado por engrenagens retas, e do pulso e cotovelo, por polias e correias de sincronização. Como os motores utilizados são DC (Direct Current), o sentido da rotação pode ser alterado ao inverter a polaridade de alimentação.

Figura 6 – Transmissão

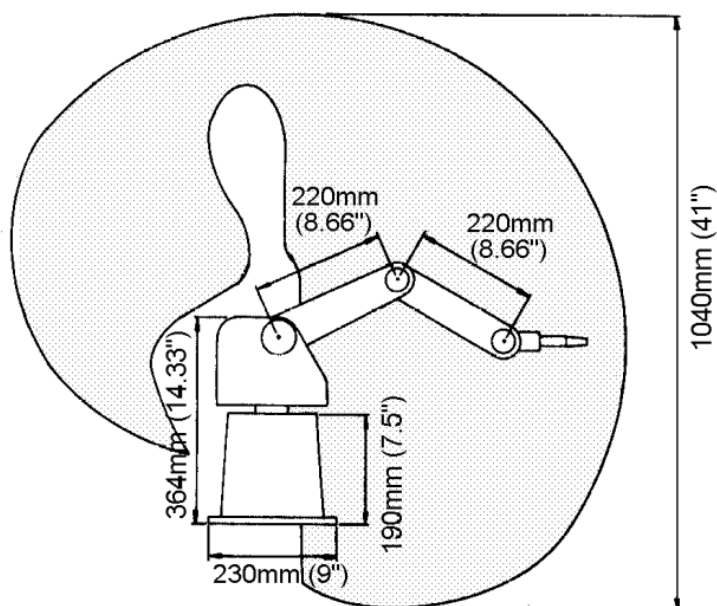


Fonte: [7].

Quando os motores 4 e 5 são acionados em direções opostas, o punho se movimenta para cima e para baixo. Ao serem acionados na mesma direção, esses motores permitem o “rolamento” do punho nos sentidos horário e anti-horário. Já a abertura e fechamento do punho é provocada por meio de um parafuso de avanço acoplado diretamente no motor 6 do robô.

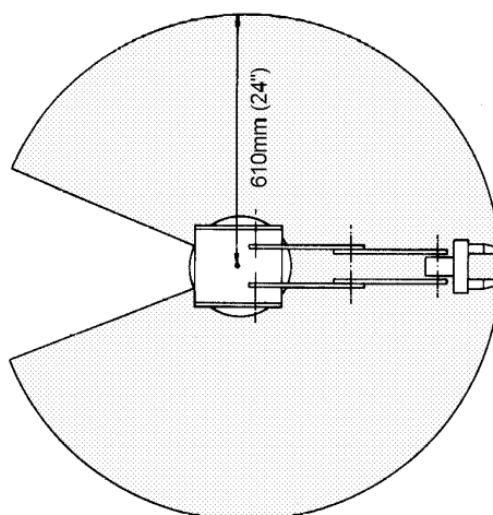
A anatomia do Scrobot proporciona um amplo espaço de trabalho devido à capacidade de rotação em torno do eixo das juntas rotativas, oferecendo maior flexibilidade de movimento e alcance. Nas Figuras 2.7 e 2.8, é possível aferir a área de trabalho desse mecanismo pelas vistas lateral e superior.

Figura 7 – Área de trabalho do Scorbot vista lateral



Fonte: [7].

Figura 8 – Área de trabalho pela vista superior



Fonte: [7].

Foram coletadas as informações mais importantes sobre as especificações técnicas do Scorbot apresentadas no manual de uso da Intelitek e estas podem ser observadas de forma resumida no Quadro 2.

Quadro 2 – Especificações técnicas do Scrobot.

Item	Nota
Estrutura mecânica	Articulada verticalmente
Número de Eixos	Cinco eixos mais um efetuador
Movimento dos Eixos:	
Eixo 1: Rotação da Base.	310°
Eixo 2: Rotação do ombro.	+130° / -35°
Eixo 3: Rotação do cotovelo.	± 130°
Eixo 4: Inclinação do punho.	± 130°
Eixo 5: Rotação do punho.	Ilimitado
Raio máximo de operação	610mm
Efetuator Final	Garra com movimento paralelo
Atuadores	12Vcc servo motores
Capacidade do motor (1-6)	70W
Transmissão	
Motores 1, 2, 3:	127.1:1
Motores 4,5:	65.5:1
Motor 6:	19.5:1
Carga máxima	1kg
Peso	11.5 kg
Velocidade máxima	600 mm/s
Temperatura de operação	2° - 40°C

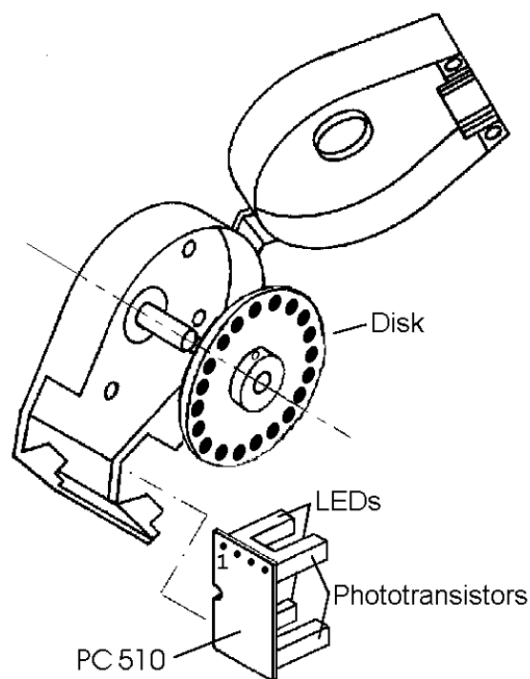
Fonte: Elaborado pela autora.

Além dos atuadores apresentados, para que o manipulador consiga desempenhar suas funções, é necessário que apresente determinados sensores responsáveis por permitir que ele interprete os comandos enviados e execute a trajetória desejada. Esses sensores são os encoders ópticos e chaves de fim de curso (também denominadas de microswitchs).

2.1.2 Encoders Ópticos

Acoplados a cada um dos seis motores do Scrobot, os encoders ópticos são compostos por um disco com padrões codificados em sua superfície e dois sensores ópticos que detectam esses padrões à medida que o disco gira. No caso do manipulador estudado nesse projeto, os sensores ópticos utilizados são fototransistores. Essa estrutura descrita pode ser observada pela Figura 9.

Figura 9 – Encoder



Fonte: [7].

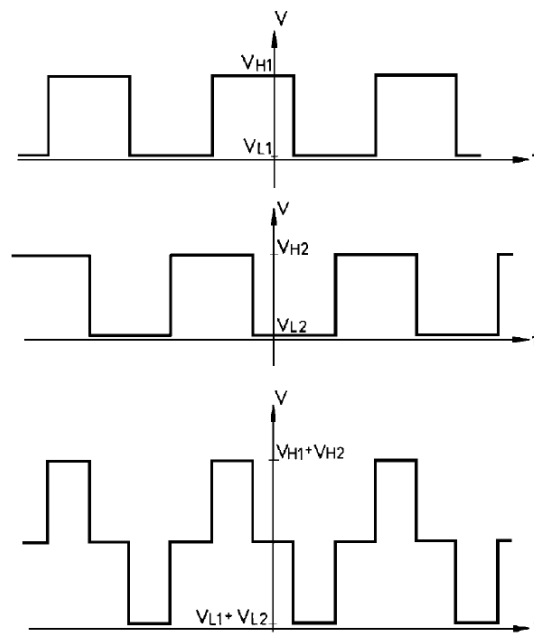
Quando o eixo do motor se move, o disco do codificador também gira. Como pode ser aferido pela Figura 9, o disco apresenta orifícios que correspondem a um padrão binário específico. Com o LED (*Light-Emitting Diode*) sempre emitindo luz, à medida que o eixo do motor se movimenta, esses sinais luminosos são captados pelo disco e convertidos em pulsos elétricos pelo fototransistor que emite níveis altos ao receber essa incidência luminosa e baixos quando não a recebe.

Contudo, o sinal gerado pelo fototransistor é relativamente fraco e apresenta baixa amplitude, sendo necessária a utilização de um circuito amplificador, no caso, o PC150. Essas informações são enviadas para o controlador do robô, que utiliza esses dados para controlar o

posicionamento dos motores na medida em que o número de sinais captados é proporcional ao quanto o motor se movimentou.

As conexões dos encoders ocorrem por meio de quatro fios: GND (terra), alimentação do LED e os pinos P0 e P1, que são responsáveis por receber os sinais captados pelos fototransistores. O diagrama da Figura 10 representa os sinais recebidos pelos fototransistores.

Figura 10 – Sinais captados pelo Encoder



Fonte: [7].

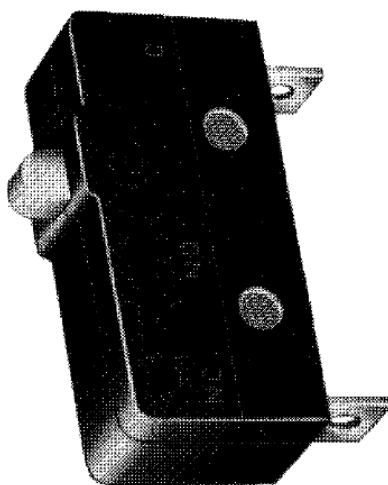
Os dados apresentados pela Figura 10 devem-se ao posicionamento defasado entre os fototransistores, sendo o primeiro gráfico referente ao sinal obtido por P0, o segundo por P1 e o terceiro pela soma dos dois sinais. A interpretação desses dados define o sentido da rotação do motor, pois caso o sensor da direita captar um sinal alto primeiro, o sentido de rotação do motor será horário, e caso o sensor a esquerda captar o sinal primeiro, será anti-horário. Já a contagem da variação dos pulsos ao longo do tempo, determina o deslocamento e velocidade angular.

2.1.3 Chaves de fim de curso

Existem cinco chaves de fim de curso, ou microswitchs, instaladas na estrutura do braço robótico cuja finalidade é estabelecer a posição inicial do manipulador robótico. Cada

eixo do robô apresenta um desses dispositivos que são capazes de determinar a posição inicial de cada elo do manipulador por meio de um sinal elétrico que é interrompido por um came ao acionar a chave quando a junta atinge o ângulo desejado. Esse dispositivo pode ser aferido pela Figura 11.

Figura 11 – Microswitch



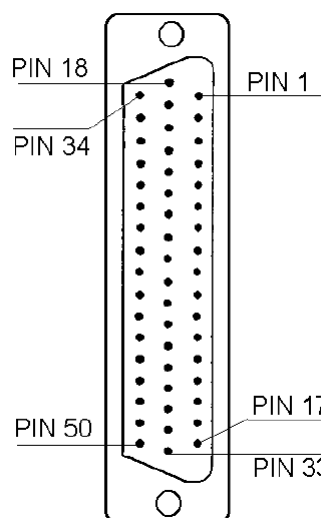
Fonte: [7].

É importante que seja desenvolvido no algoritmo de controle do manipulador uma rotina responsável por localizar a posição de referência inicial de cada junta e que esta seja realizada antes de se inicializar a trajetória definida para o manipulador.

2.1.4 Cabeamento

Os cabos dos motores no corpo do robô juntamente com os respectivos encoders e microswitches estão conectados diretamente à um conector DB-50 que pode ser aferido pela Figura 12.

Figura 12 – Conector DB-50



Fonte: [7].

São utilizadas no total 42 vias do conector, responsáveis pela alimentação e transferência de dados dos atuadores e sensores presentes no Scrobot. Na Figura 12, é possível observar a ordem em que os pinos foram enumerados. As relações entre os pinos responsáveis pelo funcionamento dos atuadores e sensores podem ser aferidas nas Tabelas 2.2, 2.3 e 2.4, referentes aos motores, encoders e chaves de fim de curso, respectivamente.

Quadro 3– Cabeamento dos motores

Eixo	Sinal	Cor da via	Pino
1	+	branco	50
	-	cinza/verde	17
2	+	branco	49
	-	branco/verde	16
3	+	branco	48
	-	laranja/marrom	15
4	+	branco	47
	-	laranja/verde	14
5	+	branco	46
	-	laranja/cinza	13
Garra	+	branco	45
	-	laranja/azul	12

Fonte: Elaborado pela Autora.

Quadro 4– Cabeamento dos Encoders

Eixo	Sinal	Cor da via	Pino
1	GND	branco	33
	P1	branco/laranja	5
	V_{LED}	amarelo	11
	P0	marrom	2
2	GND	branco	32
	P1	branco/laranja	21
	V_{LED}	amarelo	27
	P0	cinza	1
3	GND	branco	31
	P1	marrom/azul	4
	V_{LED}	amarelo	10
	P0	cinza	36
4	GND	branco	30
	P1	verde/azul	20
	V_{LED}	amarelo	26
	P0	azul	35
5	GND	branco	29
	P1	verde/azul	3
	V_{LED}	amarelo	9
	P0	azul	18
Garra	GND	branco	28
	P1	cinza/azul	19
	V_{LED}	branco	25
	P0	branco/azul	34

Fonte: Elaborado pela autora.

Quadro 5– Cabeamento dos Microswitches

Eixo	Sinal	Cor da via	Pino
1	GND	branco	33
	MS	marrom	23
2	GND	branco	32
	MS	cinza	7
3	GND	branco	31
	MS	laranja	24
4	GND	branco	30
	MS	verde	8
5	GND	branco	29
	MS	azul	6
Garra	GND	branco	28
	MS	marrom/cinza	22

Fonte: Elaborado pela autora.

Na Tabela 2.4, é possível observar que alguns pinos se repetem aos da Tabela 2.3, isso ocorre pois o GND dos microswitches e encoders foram conectados ao mesmo pino. A nomenclatura MS da Quadro 4 refere-se ao sinal do microswitch.

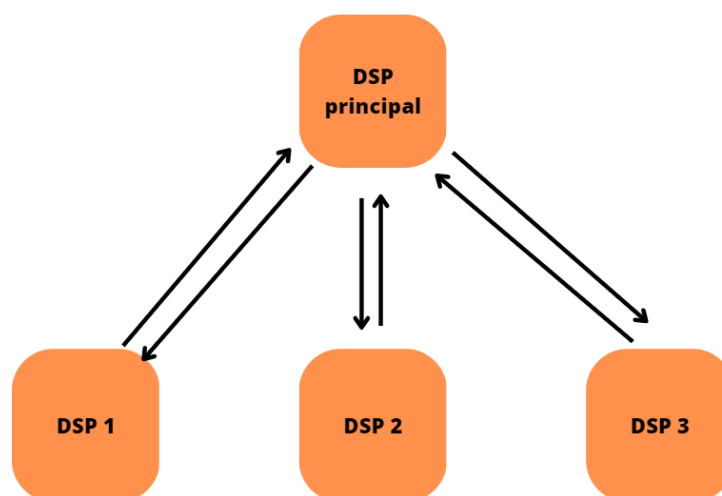
3 DESENVOLVIMENTO DA ELETRÔNICA DE CONTROLE

Essa seção tem como objetivo apresentar os recursos sugeridos para o desenvolvimento da eletrônica de controle alternativa para o manipulador ‘Scorbot-ER V+’ em uma placa de circuito impresso PCB.

3.1 IDEALIZAÇÃO DO CIRCUITO

O primeiro passo para a idealização do circuito foi determinar a implementação do CAN. Foi definida a elaboração de um modulo principal configurado para desempenhar a função de transmissão, o papel dessa DSP é de receber os dados enviados pelo computador e transmitir essas mensagens ao barramento CAN. Três outros módulos também seriam conectados ao barramento e configurados para receber dados oriundos do DSP transmissor. Um esquemático que ilustra a comunicação entre esses quatro módulos pode ser observado pela Figura 13.

Figura 13 – Esquema de comunicação entre as DSPs



Fonte: Elaborado pela autora.

Cada DSP receptor estaria encarregado de controlar dois motores do Scrobot: DSP1 controlaria o corpo e a abertura da garra, DSP2, o cotovelo e ombro e o DSP3 a inclinação e rotação do punho.

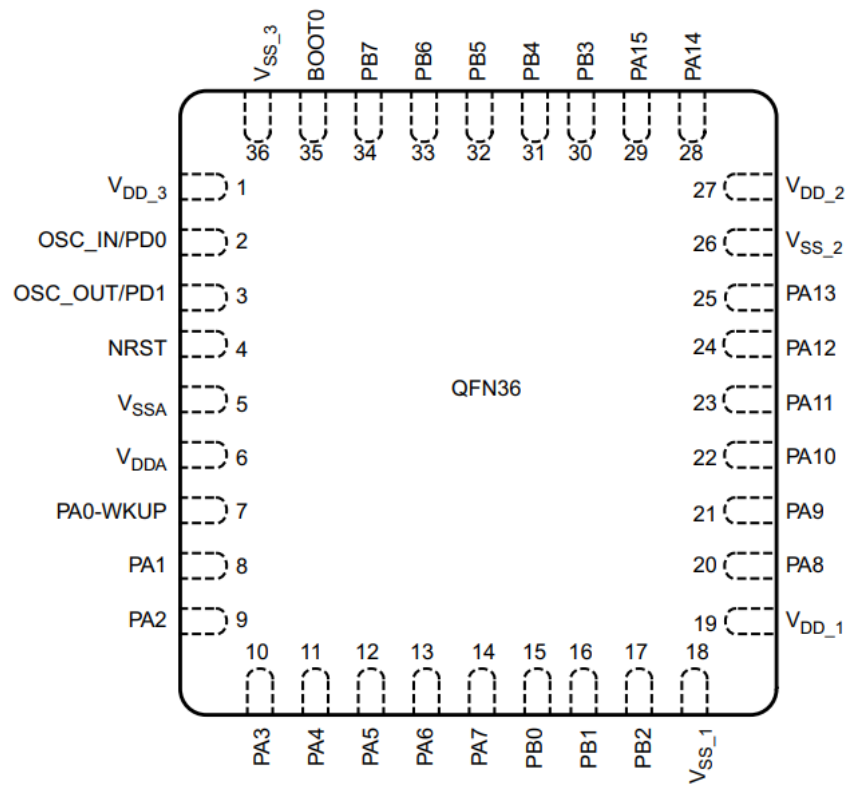
3.2 MICROCONTROLADOR

Para a escolha dos microcontroladores adequados aos DSPs receptores, foram consideradas as seguintes características:

- Número de portas de entrada/saída: o microcontrolador deve apresentar uma quantidade suficiente de portas para permitir a conexão dos sensores e atuadores.
- Poder de processamento: os módulos devem ser capazes de processar os pulsos enviados pelo encoder em tempo real, o que exige uma capacidade de processamento de alta velocidade. Também a realização de tarefas de controle envolvendo algoritmos de cinemática inversa exige um bom desempenho do processador.
- Memória: devido a capacidade do microcontrolador em armazenar dados como tabelas de cinemática, parâmetros do braço mecânico e algoritmos de controle.
- Interfaces de comunicação: é importante que o microcontrolador apresente periféricos de comunicação CAN, além de gerador de PWM, conversores AD, timers, contadores e módulos dedicados a capturar e contar pulsos
- Custo: um dos objetivos do projeto é elaborar um controlador alternativo de baixo custo.

A solução proposta para desempenhar todas as funções requeridas foi a utilização dos microcontroladores STM32 da STMicroelectronics, que possui núcleos Arm® Cortex®-M com um oscilador interno de 4-16MHz. Esse microcontrolador consta com 37 pinos GPIO, 10 conversores analógicos para digitais, integração com interfaces de comunicação moderna, como a CAN, porta USB, timers para uma execução precisa das instruções e opera em uma tensão de 3.3V. Na Figura 14 é possível observar o diagrama de pinos desse componente (COMPONENTS101, 2023).

Figura 14 – Diagrama de pinos STM32



Fonte: [9].

A função de cada pino pode ser melhor analisada pela Tabela 3.1, na qual consta as possíveis configurações de cada pino apresentado na Figura 14 (STMicroelectronics, 2022).

Quadro 6 – Configurações do diagrama de pinos do STM32

Tipo	Nome do pino	Função
Alimentação	3.3 Volts	Tensão de saída operacional
	5 Volts	Fonte de alimentação da USB ou externa
	GND	Pino de aterramento
Pinos analógicos	PA0-PA7, PB0-PB1	10 pinos ADC de resolução de 12 bits
Pinos de entrada e saída	PA0-PA15, PB0-PB15, PC13-PC15	37 pinos de entrada e saída de uso geral
Interrupções externas	PA0-PA15, PB0-PB15, PC13-PC15	Resolver problemas de temporização
PWM	PA0-PA3, PA6-PA10, PB0-PB1, PB6-PB9	15 pinos de modulação de largura de pulso

Comunicação Serial (UART)	PB6, PB7, PA2, PA3, B10, B11	Pinos RTS, CTS, USART
CAN	PA11-PA12, PB10-PB9	pinos do barramento de rede da área do controlador CAN
I2C	PB6-PB9	Dados Seriais

Fonte: Elaborado pela autora.

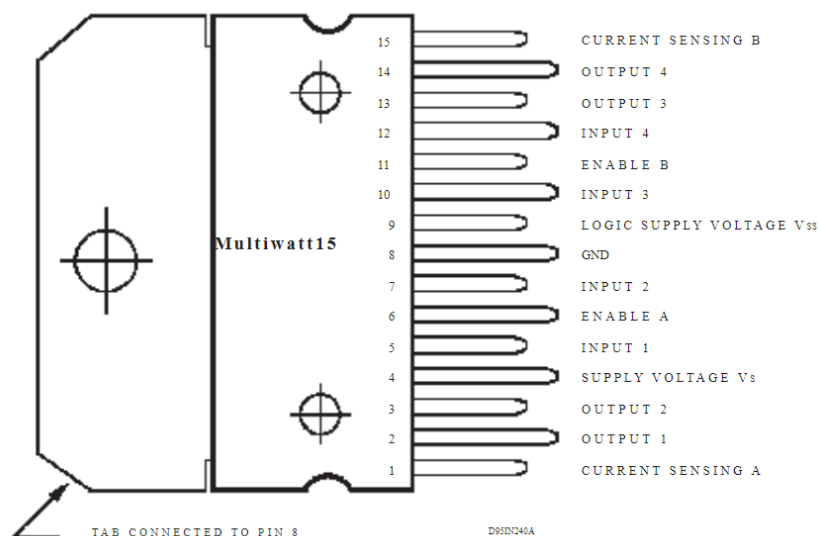
Além do microcontrolador, cada modulo deveria apresentar sensores adequados para permitir a comunicação com os atuadores, conectores e transceptor CAN, que converte os bits vindos do microcontrolador em um sinal diferencial para o barramento e vice-versa.

3.3 ESPECIFICAÇÃO DOS COMPONENTES

3.3.1 Ponte H

O acionamento dos motores controlados em cada módulo ocorre por meio de um driver L298N, que utiliza dois circuitos ponte H para controlar com eficiência a direção e velocidade de motores de corrente contínua. Esse componente é capaz de operar em uma faixa de tensão de 5V a 35V, e suporta correntes de até 2A por ponte H. Além disso, o driver pode fornecer uma tensão lógica de até 5V. Na Figura 15 é possível verificar o funcionamento do circuito desse componente (ELETROGATE, 2020).

Figura 15 – Pinos do driver L298N



Fonte: [11].

É possível destacar na Figura 15 que o driver L298N apresenta quatro interruptores controlados eletronicamente, dispostos em pares (OUTPUT:1-4). Cada par de interruptores é conectado a um terminal do motor, permitindo controlar a direção do fluxo de corrente. Ao alternar os interruptores, é possível inverter a polaridade do sinal de saída, assim, a polaridade aplicada ao motor é invertida, permitindo o controle bidirecional deste.

Para controlar a velocidade dos motores, o driver L298N utiliza um sinal PWM (*Pulse Width Modulation*). O PWM é uma técnica que consiste em modular a largura de um pulso de sinal em intervalos regulares. Ao variar a largura dos pulsos, é possível controlar a média de tensão entregue ao motor, resultando em uma variação proporcional da velocidade. No projeto, o sinal PWM é gerado pelo STM32 e enviado ao driver L298N, que o interpreta e ajusta a tensão fornecida aos motores de acordo com a largura do pulso (MARZOCCA, 2022).

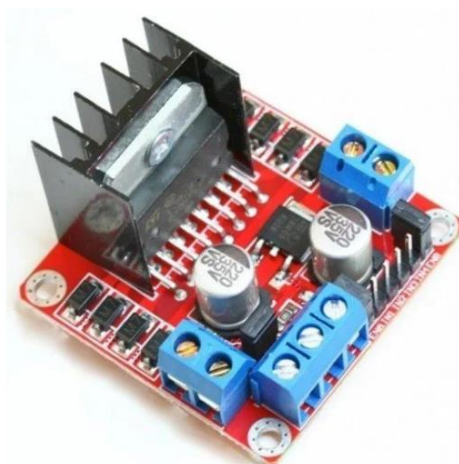
O STM32 inclui quinze pinos capazes de gerar PWM, sendo que cada pino apresenta uma resolução de 16 bits, logo, o contador do timer pode variar em 2^{16} intervalos (0 a 65535). Como foi explicado, esse sinal é emitido por uma porta digital do microcontrolador, alternando os níveis lógicos alto e baixo em uma frequência específica. Assim, um parâmetro fundamental no PWM é o ciclo de trabalho, que representa a proporção de tempo em que o sinal está no estado alto em relação ao período completo do pulso (COMPONENTS101, 2023).

De forma prática, o controle da velocidade ocorre alterando o ciclo de trabalho. Um ciclo de trabalho de 50% é obtido quando o valor do contador definido na programação do STM

for ajustado para 32767, reduzindo pela metade a tensão fornecida ao motor. Já um ciclo de 100% forneceria a tensão máxima, logo, a rotação do motor apresentaria o valor máximo.

É possível conferir o driver utilizado no projeto na Figura 16. Em cada um dos três DSP receptores foi acoplada uma ponte H, sendo que esta é capaz de controlar 2 motores de corrente contínua simultaneamente.

Figura 16 – Driver L298N



Fonte: [10].

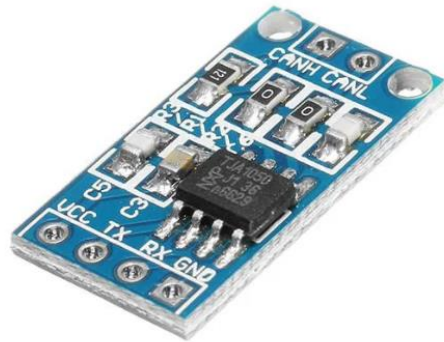
3.3.2 Transceptor TJA1050

Entre os periféricos disponíveis pelo STM32, existe um controlador CAN integrado à placa, responsável por processar os dados de transmissão (Tx) e de recepção (Rx) fornecendo uma interface lógica com o CAN transceptor. Os detalhes dessa arquitetura foram abordados na seção 2 e podem ser observados pela Figura 3.

O papel do CAN transceptor nesse contexto é fornecer uma interface elétrica entre o controlador CAN e o barramento, convertendo o dado em sinal elétrico e vice-versa. Dessa forma, o transceptor CAN permite a comunicação bidirecional de dados entre o microcontrolador e os dispositivos conectados ao barramento CAN.

Foi escolhido para o projeto o módulo serial CAN BUS TJA1050, devido ao baixo custo e fácil acesso. Esse componente pode ser observado na Figura 17.

Figura 17 – Módulo CAN BUS TJA1050

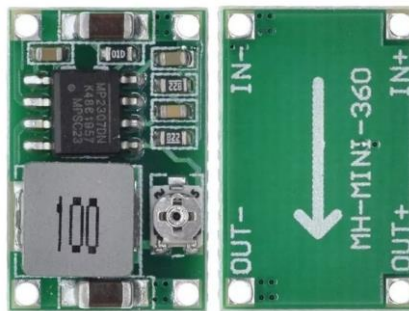


Fonte: [13].

3.3.3 Conversor DC-DC Step Down

Cada módulo dos DSP 1, 2 e 3 apresentaria uma alimentação geral de 12V, com o intuito de fornecer a tensão adequada aos motores do robô. Contudo, os demais módulos citados nas seções anteriores operam entre 3.3 e 5V. A solução elaborada para contornar esse problema foi a utilização do regulador de tensão Mini 360, que pode ser aferido pela Figura 18.

18 – Regulador de tensão Mini 360



Fonte: [14].

O componente da Figura 18 é responsável por converter uma tensão de entrada mais alta para uma tensão de saída estável e regulada, por meio do armazenamento e liberação de energia por um indutor interno, controlado por um transistor de comutação. A regulação da tensão de saída é realizada ajustando o potenciômetro presente no módulo, oferecendo uma solução simples e barata para o problema.

3.3.4 Conectores

A escolha dos conectores foi projetada considerando a tensão e corrente que cada um desses elementos estaria submetido. Para o barramento da alimentação geral e dos motores, como é necessário que operem a altas tensões, foram utilizados bornes de 2 vias por segurança, sendo que, de acordo com as especificações técnicas desses (ELETROGATE), suportam uma tensão máxima de 300V e corrente de até 10A.

Já para as demais conexões da placa, por trabalharem com tensões de 3.3 à 5V, foram selecionadas barras de pinos para essa função. Também foi adquirido um conector Db50 fêmea, com o qual poderiam ser soldados os fios que estabeleceriam as conexões entre o controlador alternativo e o manipulador robótico. Os conectores citados podem ser verificados na Figura 19.

Figura 19 – Conectores



Fonte: Elaborado pela autora.

3.4 MÓDULO DE CONTROLE PRINCIPAL

Idealizando projetos futuros sobre a criação de um aplicativo para controle remoto do manipulador, foi optada a utilização de um microcontrolador alternativo para o módulo emissor, sendo esse o ESP32. Desenvolvido pela empresa chinesa Espressif, o ESP32 apresenta 34

portas programáveis GPIOs, o que confere a esse componente uma ampla expansibilidade, interface CAN e conectividade embutida para Wi-Fi e Bluetooth, sendo mais conveniente para elaboração de um projeto que necessite de comunicação sem fio.

Outra vantagem na utilização dessa placa é que ela, assim como o STM32, pode ser programada utilizando a IDE do Arduino, portanto, o desenvolvimento da lógica do programa poderia ser realizado com a mesma interface em todos os módulos por meio da linguagem C/C++.

4 RESULTADOS E DISCUSSÕES

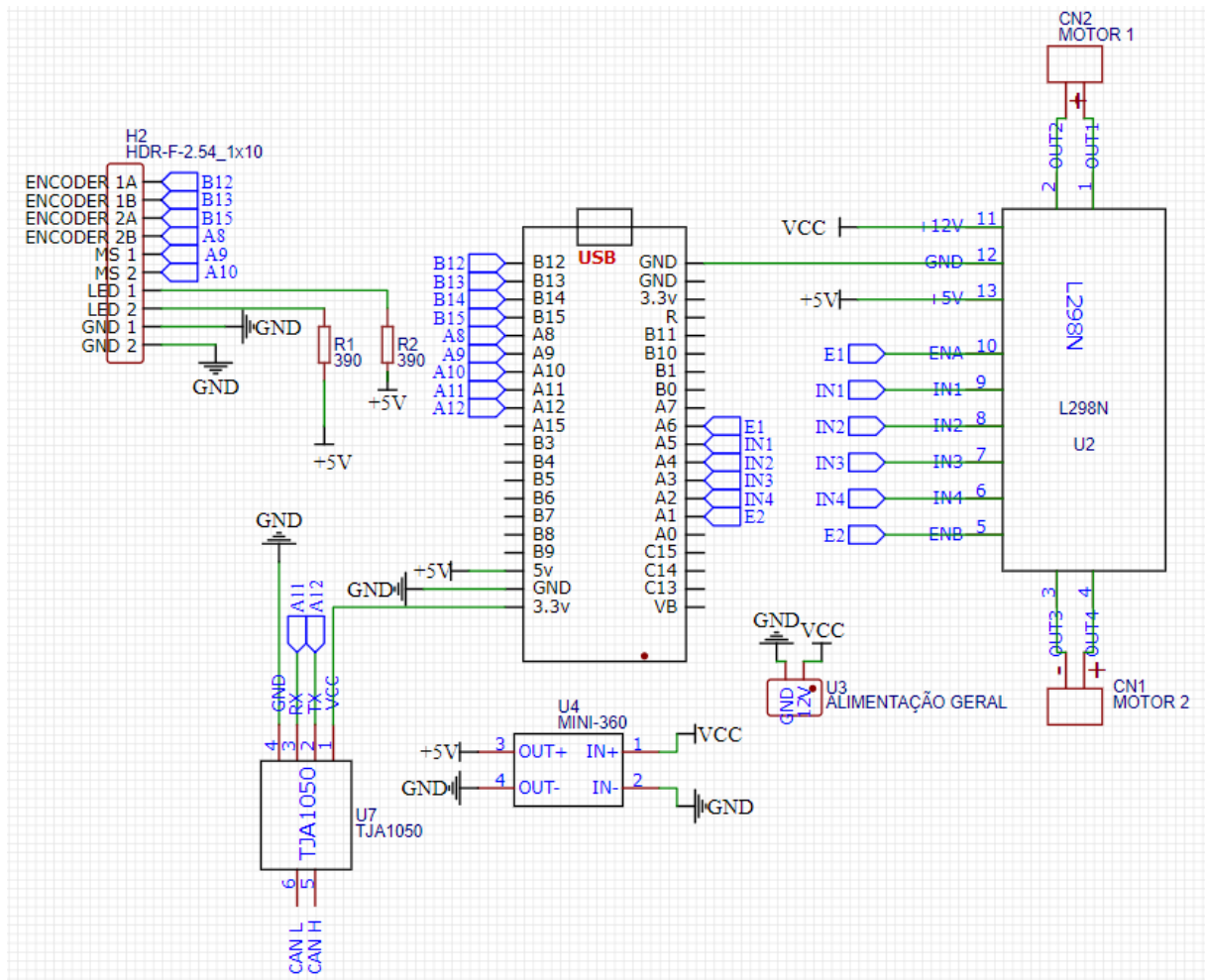
4.1 CONFECÇÃO DOS MÓDULOS DE CONTROLE DOS ATUADORES

O desenvolvimento do circuito e elaboração da PCB foi realizado utilizando o software EasyEDA, sendo essa uma ferramenta gratuita de design eletrônico. Utilizando as especificações técnicas apresentadas na seção 3, foram elaboradas as conexões entre os componentes necessários a cada módulo de controle dos atuadores do Scorbob. O esquemático dessas conexões pode ser aferido pela Figura 20.

Objetivando uma maior organização e melhor entendimento, foram utilizadas as ferramentas de ligações do tipo Porta-Net na Figura 20, evitando poluir o esquemático com um excesso de fios.

Já na Figura 21, é possível aferir o layout produzido para PCB por meio do esquemático elaborado. Foi utilizada a ferramenta de roteamento automático do software selecionado e alteradas algumas configurações para atender as necessidades do projeto, tais como a largura da trilha definida de 1,5 mm com afastamento de 0,22 mm. Também os Pads de cada componente foram aumentados em, pelo menos, 0,1mm objetivando facilitar o processo de solda.

Figura 20 – Esquemático das conexões dos módulos de controle dos atuadores



Fonte: Elaborado pela autora.

O papel da barra de 10 pinos, nomeada “H2” no esquema da Figura 21, é estabelecer as conexões entre o robô e o microcontrolador, apresentando quatro pinos dedicados a leitura dos encoders, dois à verificação do sinal das chaves de fim de curso, dois para a alimentação dos leds e dois para aterramento. Ao definir os pinos do controlador, existiu uma preocupação em selecionar portas lógicas que operassem com interrupção, no caso dos encoders.

Os outros dois bornes são responsáveis por fornecer a tensão da ponte H para o acionamento dos motores.

Na Quadro 7 pode ser observada a disposição dos pinos selecionados no microcontrolador com suas respectivas funções.

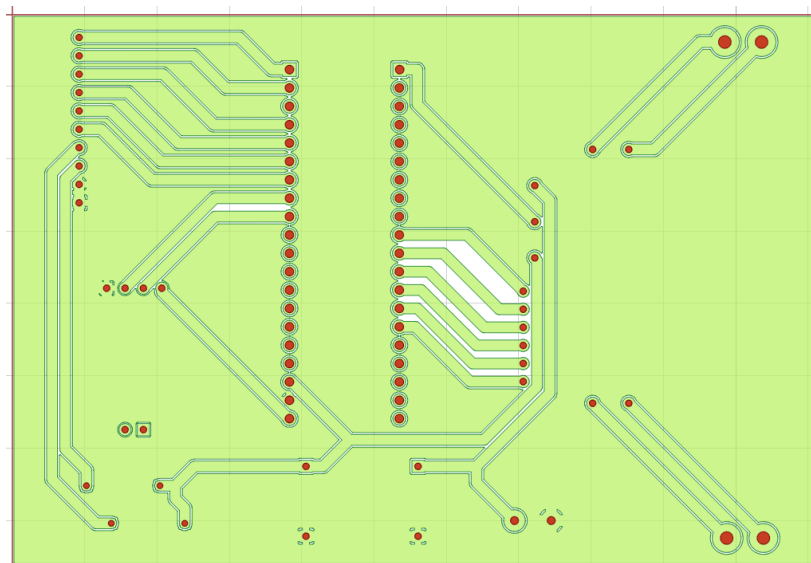
Quadro 7 – Funções dos pinos do STM32

Pino	Tipo	Função
B12	Entrada digital	Sinal P0 do encoder 1
B13	Entrada digital	Sinal P2 do encoder 1
B15	Entrada digital	Sinal P0 do encoder 2
A8	Entrada digital	Sinal P2 do encoder 2
A9	Entrada Digital	Sinal da chave de fim de curso 1
A10	Entrada Digital	Sinal da chave de fim de curso 2
A11	Comunicação	RX do TJA1050
A12	Comunicação	TX do TJA1050
A6	Saída PWM	Ativação A da ponte H
A5	Saída digital	Entrada 1 da ponte H
A4	Saída digital	Entrada 2 da ponte H
A3	Saída digital	Entrada 3 da ponte H
A2	Saída digital	Entrada 4 da ponte H
A1	Saída PWM	Ativação B da ponte H

Fonte: Elaborado pela autora.

A próxima etapa do projeto consistiu na confecção do circuito em uma placa de fenoline. Assim, o arquivo gerado pelo easyEDA foi transferido para o software de fabricação de circuitos impressos FlatCAM, permitindo converter o arquivo Gerber em uma trajetória utilizada por uma máquina CNC para criar as trilhas e furos necessários no PCB. Na Figura 22 é possível aferir o resultado obtido por meio dessa etapa.

Figura 22 – Trajetória da CNC especificada no FlatCAM

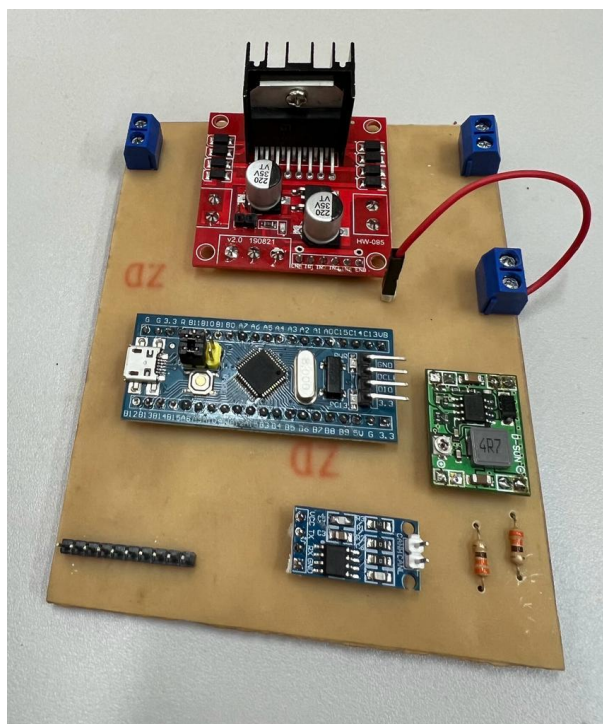


Fonte: Elaborado pela autora.

É importante ressaltar que, para a confecção da placa na máquina CNC o arquivo gerado pelo FlatCAM esteja posicionado nas coordenadas iniciais X: 0 e Y:0, facilitando o trabalho de posicionar a placa de fenolite adequadamente na mesa da CNC.

A placa obtida por meio dessas etapas pode ser aferida na Figura 23, apresentando o circuito final de testes dos atuadores do Scrobot.

Figura 23 – Confecção dos módulos de controle dos atuadores para testes



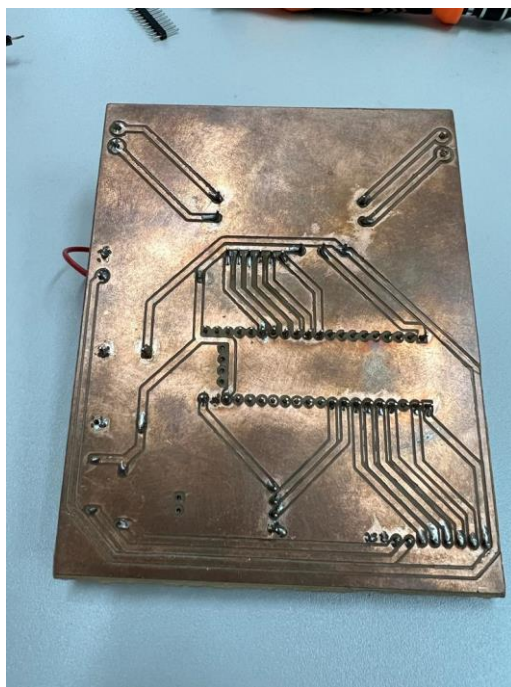
Fonte: Elaborado pela autora.

Devido a uma falha no dimensionamento do cronograma do projeto, não foi possível testar o módulo diretamente no Scrobot, sendo realizado apenas um teste para acionamento de motores DC de 6V acoplados a um encoder óptico. Como o acionamento dos atuadores do Scrobot apresentam a mesma lógica utilizada, diferindo apenas no número de pulsos dos encoders e voltagem do motor DC, o teste forneceu uma verificação do funcionamento esperado pelo circuito elaborado.

Utilizando um multímetro, foi possível verificar as ligações entre os componentes por meio de um teste de impedância após o processo de solda. Também com auxílio de uma fonte de 12V, foi possível realizar o teste da placa de circuito impresso e conferir as tensões presentes

na trilha para garantir o funcionamento correto dos componentes por meio de um multímetro. Na Figura 24 é possível aferir a solda realizada na placa.

Figura 24 – Solda realizada na placa de testes



Fonte: Elaborado pela autora.

Foi adicionada uma ilha na placa, como pode ser observado pelas Figuras 22 e 23, com o objetivo de permitir a execução dos testes enquanto o microcontrolador é conectado via USB ao computador, para evitar que este seja alimentado por duas vias diferentes. Ao ser testado apenas pela fonte externa, o cabo vermelho pode ser removido.

4.2 TRABALHOS FUTUROS

Com as referências coletadas ao longo do projeto e a confecção de módulos para teste do controle dos atuadores do Scorbobot, as próximas etapas que podem ser desenvolvidas em trabalhos futuros seriam:

- Implementação do módulo no controlador do Scorbobot e verificação dos pulsos obtidos por meio dos encoders do manipulador para conseguir converter a quantidade de pulsos que equivalem a determinados graus, sendo que essa relação

é diretamente proporcional.

- Implementação de uma rotina “home” na qual o manipulador acionaria cada motor até todos os eixos se encontrarem na posição de referência inicial, testando, assim, o funcionamento das chaves de fim de curso.
- Implementação de algoritmos de controle mais refinados que utilizam dos conceitos de cinemática direta e inversa para obter a trajetória do efetuador.

É possível enfatizar também que, a utilização do barramento CAN como método para comunicação entre os módulos de controle atribui ao projeto grande versatilidade, assim, esse controlador poderia ser aproveitado em qualquer projeto que tenha como objetivo o controle de motores DC, adicionando ou removendo com facilidade os módulos que seriam utilizados dependendo da quantidade de eixos necessários para o controle. Pode ser citado como exemplo dessa aplicação a utilização dos módulos elaborados em uma Router CNC, sendo que essa máquina conta com motores DC e chaves de fim de curso para realizar seu funcionamento.

5 CONCLUSÃO

Existe uma demanda crescente de profissionais especializados na área de robótica que estejam familiarizados com a utilização de sistemas de comunicação modernos, linguagem de programação e automação. O desenvolvimento do protótipo proposto pode ser utilizado como uma ferramenta valiosa para o desenvolvimento dessas habilidades.

A implementação do Scorbot com o controlador de arquitetura aberta nas salas de aula permite uma análise prática e aprofundada dos conceitos teóricos estudados, permitindo aos alunos explorar de forma concreta os princípios da cinemática, controle de movimento, programação de robôs e integração de sistemas, além de compreenderem os desafios e as considerações práticas envolvidas na operação de um robô industrial. Também estimula um estudo da linguagem de programação C, que é amplamente utilizada em aplicações robóticas nas indústrias.

Além disso, a utilização do protocolo CAN proporciona aos estudantes uma experiência direta com uma tecnologia amplamente utilizada nas indústrias.

É importante ressaltar também que esse projeto viabiliza a utilização do Scorbot que se encontra no laboratório da instituição e não é aproveitado pela faculdade devido ao funcionamento desatualizado do controlador atual. A implementação do controlador alternativo objetiva aprimora-lo e torna-lo mais versátil.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- [1]: SAHA, S. K. Introducción a la Robótica. Noida: McGraw-Hill, 2008.
- [2]: SCHLEGEL, Christian. The role of CAN in the age of Ethernet and IoT. 16th INTERNATIONAL CAN CONFERENCE (ICC), 2017, Nuremberg, Germany.
- [3]: Di Natale, M. (2008). Understanding and using the Controller Area Network.
- [4]: SCHÄUFFELE, J. Automotive Software Engineering: Principles, Processes, Methods, and Tools. 2005.
- [5]: NAVET, Nicolas; SIMONOT-LION, Françoise. Automotive Embedded Systems Handbook. Estados Unidos: CRC Press, 2008.
- [6]: TEXAS INSTRUMENTS. Introduction to the Controller Area Network (CAN) Application Report. SLOA101B. Dallas: Texas Instruments, agosto de 2002. Revisado em maio de 2016. Autor: Steve Corrigan. Título da seção: Industrial Interface.
- [7]: Intelitek Inc. (2003). SCORBOT-ER 5Plus User Manual. Catálogo #100016, Rev. C. Copyright 2003 Intelitek Inc.
- [8]: COMPONENTS101 - STM32F103C8T6 - Blue Pill Development Board. Disponível em < <https://components101.com/microcontrollers/stm32f103c8t8-blue-pill-development-board>>. Acesso em 01 mai. 2023.
- [9]: STMicroelectronics. (2022). STM32F103x8, STM32F103xB Datasheet. DS5319 Rev 18. STMicroelectronics. Obtido do arquivo PDF.
- [10]: ELETROGATE. (2020). Guia definitivo de uso da Ponte H L298N. Disponível em < <https://blog.eletrogate.com/guia-definitivo-de-uso-da-ponte-h-l298n/>> Acesso em 01 mai. 2023.
- [11]: STMicroelectronics. (2000). L298HN - Dual Full Bridge Driver Datasheet. Reduparado de: <http://www.st.com>
- [12]: Pop, A.A. Aquisição de Velocidade do Encoder Incremental Utilizando um Microcontrolador STM32 e o NI ELVIS. Sensores 2022, 22, 5127. <https://doi.org/10.3390/s22145127>. Editor Acadêmico: Cristoforo Marzocca. Recebido: 11 de maio de 2022; Aceito: 4 de julho de 2022; Publicado: 8 de julho de 2022.
- [13]: NXP Semiconductors. TJA1050 - CAN Bus Transceiver. Disponível em: <https://www.nxp.com/products/interfaces/can-transceivers/can-lin-transceivers/tja1050-can-bus-transceiver:TJA1050>. Acesso em: 5 de jun. de 2023.
- [14]: Arduino e Cia. Regulador de Tensão Mini 360. Disponível em: <https://www.arduino-e-cia.com.br/regulador-de-tensao-mini-360/>. Acesso em: 5 de jun de 2023.