



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIAS E TECNOLOGIA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

MARCOS ROBERTO EDUARDO DE ALBUQUERQUE

**DESENVOLVIMENTO DE UM SISTEMA EMBARCADO PARA CONTROLE
MANUAL E AUTOMÁTICO DE UM MANIPULADOR ROBÓTICO**

PAU DOS FERROS - RN

2023

MARCOS ROBERTO EDUARDO DE ALBUQUERQUE

**DESENVOLVIMENTO DE UM SISTEMA EMBARCADO PARA CONTROLE
MANUAL E AUTOMÁTICO DE UM MANIPULADOR ROBÓTICO**

Monografia apresentada à Universidade Federal Rural do Semi-Árido – UFERSA, Campus Pau dos Ferros, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Profº Dr. Cecilio Martins de Sousa Neto.

PAU DOS FERROS - RN

2023

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

AA345 d Albuquerque, Marcos Roberto Eduardo de.
Desenvolvimento de um sistema embarcado para controle manual e automático de um manipulador robótico / Marcos Roberto Eduardo de Albuquerque. - 2023.
37 f. : il.

Orientador: Cecilio Martins de Sousa Neto.
Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de Engenharia de Computação, 2023.

1. Automação. 2. Robótica. 3. Microcontrolador. 4. Sistema embarcado. I. Neto, Cecilio Martins de Sousa, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade com AACR2 e os dados fornecidos pelo(a) autor(a).

Biblioteca Campus Pau dos Ferros
Bibliotecária: Erlanda Maria Lopes da Silva
CRB: 15/966

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

MARCOS ROBERTO EDUARDO DE ALBUQUERQUE

**DESENVOLVIMENTO DE UM SISTEMA EMBARCADO PARA CONTROLE
MANUAL E AUTOMÁTICO DE UM MANIPULADOR ROBÓTICO**

Monografia apresentada à Universidade Federal Rural do Semi-Árido – UFERSA, Campus Pau dos Ferros, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

Defendida em: 04 / 10 / 2023.

BANCA EXAMINADORA

Cecilio Martins de Sousa Neto, Prof. Dr. (UFERSA)
Orientador

Francisco Carlos Gurgel da Silva Segundo, Prof. Dr. (UFERSA)
Membro Examinador

Petrúcio Ricardo Tavares de Medeiros, Prof. Dr. (UFERSA)
Membro Examinador

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão à minha família pelo apoio incondicional ao longo de minha jornada educacional. Seu amor, incentivo e compreensão foram fundamentais para a minha busca pelo conhecimento e minha realização pessoal.

Agradeço de coração aos meus professores, cuja orientação e sabedoria moldaram meu pensamento e meu crescimento acadêmico. Suas lições e inspiração desempenharam um papel essencial na formação deste trabalho.

Aos meus queridos amigos, agradeço por compartilharem as alegrias e desafios desta jornada acadêmica. Suas palavras de encorajamento e apoio moral foram um alicerce essencial para superar obstáculos e alcançar este marco.

Este trabalho não teria sido possível sem o apoio, a orientação e o incentivo de todos vocês. Agradeço do fundo do meu coração por fazerem parte da minha jornada e por contribuírem para o meu crescimento pessoal e profissional.

Se alguém pode refutar-me e provar de modo conclusivo que penso ou procedo incorretamente, de bom grado mudarei minha forma de agir. Pois persigo a verdade, que nunca prejudicou ninguém; ao contrário, prejudica-se o que persiste em seu próprio engano ou ignorância.

– Marco Aurélio

RESUMO

Este trabalho aborda o desenvolvimento de um sistema embarcado para controle manual e automático de um manipulador robótico utilizando o microcontrolador ATmega2560. A motivação deste trabalho advém da aplicação prática dos conhecimentos de Engenharia de Computação, a exploração deste microcontrolador específico e a potencial utilidade do robô como uma ferramenta prática de laboratório. O principal objetivo deste trabalho é desenvolver um sistema embarcado para controle manual e automático de um manipulador robótico. Especificamente, pretende-se programar o ATmega2560 em linguagem C para controlar o manipulador, bem como desenvolver um sistema que grava e reproduz trajetórias de forma automática. O sistema foi desenvolvido em dois modos: manual e automático. Utilizou-se a modulação por largura de pulso (PWM) para controlar os servos e a conversão analógico-digital para ler valores dos potenciômetros. O firmware foi desenvolvido diretamente para o ATmega2560, sem camadas de abstração, e utilizou-se a biblioteca "avr/eeprom.h" para armazenar informações das trajetórias na EEPROM. O manipulador robótico foi testado com êxito, alcançando resultados promissores no controle manual e automático. Os desafios, como instabilidade e limitações da memória EEPROM, foram identificados. Porém, as soluções, como a integração de memórias externas e o uso de PCBs adequadas, sugerem vias claras para melhorias futuras e otimizações.

Palavras-chave: Automação. Robótica. Microcontrolador. Sistema embarcado.

ABSTRACT

This work deals with the development of an embedded system for manual and automatic control of a robotic manipulator using the ATmega2560 microcontroller. The motivation for this work comes from the practical application of Computer Engineering knowledge, the exploration of this specific microcontroller and the potential usefulness of the robot as a practical laboratory tool. The main objective of this work is to develop an embedded system for manual and automatic control of a robotic manipulator. Specifically, we intend to program the ATmega2560 in C language to control the manipulator, as well as develop a system that records and reproduces trajectories automatically. The system was developed in two modes: manual and automatic. Pulse width modulation (PWM) was used to control the servos and analog-to-digital conversion to read potentiometer values. The firmware was developed directly for the ATmega2560, without abstraction layers, and the "avr/eeprom.h" library was used to store trajectory information in the EEPROM. The robotic manipulator was successfully tested, achieving promising results in manual and automatic control. Challenges such as instability and limitations of EEPROM memory have been identified. However, solutions, such as the integration of external memories and the use of suitable PCBs, suggest clear paths for future improvements and optimizations.

Keywords: Automation. Robotics. Microcontroller. Embedded system.

LISTA DE FIGURAS

Figura 1	– Ilustração de um manipulador robótico do tipo cartesiano	21
Figura 2	– Ilustração de um manipulador robótico do tipo cilíndrico	21
Figura 3	– Ilustração de um manipulador robótico do tipo esférico	22
Figura 4	– Ilustração de um manipulador robótico do tipo SCARA	23
Figura 5	– Ilustração de um manipulador robótico do tipo antropomórfico	23
Figura 6	– Diagrama básico da arquitetura de um microcontrolador	24
Figura 7	– Diagrama da arquitetura do ATmega640/V-1280/V-1281/V-2560/V-2561/V ..	26
Figura 8	– Servo motor	29
Figura 9	– Sinal PWM para o controle do servo motor	29
Figura 10	– Arquitetura do sistema	31
Figura 11	– Visão geral do sistema	37
Figura 12	– Controle manual do manipulador robótico	38
Figura 13	– Leitura do sinal PWM gerado pelo ATmega2560	38

LISTA DE TABELAS

Tabela 1	–	Descrição do bit de seleção do <i>clock</i>	33
Tabela 2	–	Descrição do bit do modo de geração de forma de onda	34

SUMÁRIO

1 INTRODUÇÃO.....	10
1.1 JUSTIFICATIVA.....	11
1.2 OBJETIVOS.....	13
2 ESTADO DA ARTE.....	14
2.1 TIPOS DE MANIPULADORES.....	14
2.1.1 Cartesiano.....	14
2.1.2 Cilíndrico.....	15
2.1.3 Esférico.....	16
2.1.4 SCARA.....	16
2.1.5 Antropomórfico.....	17
2.2 O MICROCONTROLADOR ATmega2560.....	18
2.2.1 EEPROM.....	21
2.2.1 Modulação por Largura de Pulso (PWM).....	21
2.4 SERVO MOTOR.....	22
3 TRABALHOS RELACIONADOS.....	24
4 PROJETO DO MANIPULADOR ROBÓTICO.....	25
4.1 ARQUITETURA GERAL DO SISTEMA.....	25
4.2 GERAÇÃO DE SINAL PWM.....	26
4.2.1 Configuração do valor de Prescaler.....	27
4.2.2 Configuração dos Timers/Contadores.....	27
4.3 DESENVOLVIMENTO DO FIRMWARE.....	29
5 RESULTADOS E DISCUSSÕES.....	31
5.1 CONTROLE MANUAL.....	31
5.2 MODO AUTOMÁTICO.....	33
6 CONSIDERAÇÕES FINAIS.....	35
REFERÊNCIAS.....	36

1 INTRODUÇÃO

A automação é a tecnologia que utiliza sistemas mecânicos, eletroeletrônicos e computacionais para operar e controlar a produção. Isso se reflete em exemplos como linhas de produção industrial, máquinas de montagem automatizadas, sistemas de controle de produção com *feedback*, máquinas-ferramentas controladas numericamente e robôs industriais (Bayer *et al*, 2011). A automação destaca-se na aplicação de manipuladores robóticos para automatizar tarefas repetitivas, além de ser uma ferramenta valiosa no ensino de robótica, permitindo a construção personalizada desses robôs através da impressão 3D, tornando o aprendizado mais prático e acessível às necessidades específicas de cada projeto.

A robótica, por sua vez, é o ramo da engenharia que se dedica ao projeto, construção e operação de máquinas capazes de realizar tarefas físicas de forma autônoma ou semi-autônoma (Ottoni, 2010). Desta forma, o objetivo da robótica é melhorar a produtividade, segurança e qualidade de vida das pessoas, além de explorar novas fronteiras tecnológicas, combinando diversos campos, como inteligência artificial e controle.

Com o avanço tecnológico, surgiram diversos tipos de manipuladores robóticos que são capazes de emular o movimento do braço humano. Esses manipuladores são amplamente utilizados em diferentes processos industriais. Eles são compostos por componentes essenciais, como juntas, elos e garras. Essas partes trabalham em conjunto para fornecer ao robô a capacidade de realizar uma ampla variedade de tarefas. Os manipuladores robóticos têm desempenhado um papel importante na automação industrial, melhorando a eficiência e a precisão dos processos de produção (Simplicio, 2016).

As aplicações dos manipuladores robóticos na indústria são diversas e dependem do tipo de tarefa, do produto e do ambiente. Alguns exemplos de aplicações são: montagem, soldagem, pintura, transporte, cirurgias de precisão, além de poderem trabalhar em ambientes perigosos que envolvem altas tensões, mineração e metalurgia (Cocota, 2013).

As características construtivas dos manipuladores robóticos envolvem aspectos como a estrutura mecânica, o sistema de acionamento, o sistema de controle e o sistema de sensoramento. A estrutura mecânica é formada por uma base fixa e uma série de elos articulados que permitem o movimento do manipulador. O número e o tipo de articulações determinam os graus de liberdade (*degrees-of-freedom* - DOF) do manipulador, ou seja, o número de parâmetros independentes que definem sua configuração espacial. Os DOF

influenciam na capacidade do manipulador de alcançar diferentes posições e orientações no espaço de trabalho. Na extremidade do manipulador, há uma ferramenta chamada *end-effector*, que pode ser uma garra, um copo de sucção, uma pistola de solda, uma câmera, etc. O *end-effector* é responsável por interagir com o objeto ou o ambiente (Santos, 2003).

Dentro desse contexto, o objetivo primordial deste trabalho é desenvolver um sistema embarcado para controle manual e automático de um manipulador robótico utilizando o microcontrolador ATmega2560. O propósito é capacitar o manipulador a realizar movimentos autônomos e precisos, alinhando-se com os avanços na automação de tarefas repetitivas e na aplicação da robótica.

O presente trabalho está organizado da seguinte forma: no Capítulo 2, serão abordados aspectos relacionados aos tipos de manipuladores robóticos existentes, uma análise do microcontrolador ATmega2560 e uma explanação sobre os servomotores utilizados no projeto. No Capítulo 3, serão apresentados os trabalhos relacionados, fornecendo um panorama das pesquisas e avanços recentes na área de controle para manipuladores robóticos. O Capítulo 4 discutirá a metodologia empregada, descrevendo o processo de desenvolvimento do sistema e suas etapas. No Capítulo 5, serão apresentados os resultados obtidos e as discussões pertinentes, destacando tanto os sucessos quanto os desafios enfrentados durante a implementação. Por fim, no Capítulo 6, serão apresentadas as considerações finais, ressaltando a relevância e as contribuições deste trabalho, bem como sugestões para trabalhos futuros na área de automação e robótica.

1.1 JUSTIFICATIVA

O avanço da automação e robótica tem desempenhado um papel crucial em diversas áreas, desde a indústria até a medicina. A utilização de robôs industriais e braços robóticos tem se tornado cada vez mais comum, visto que eles oferecem maior precisão e eficiência em tarefas complexas e repetitivas. A engenharia de computação desempenha um papel fundamental no desenvolvimento desses sistemas, unindo os princípios da engenharia elétrica e da ciência da computação para criar soluções inovadoras.

Neste trabalho, a proposta de desenvolver um sistema embarcado para controle manual e automático de um braço robótico utilizando o microcontrolador ATmega2560 é altamente relevante e motivada pelas seguintes razões:

- **Aplicação Prática dos Conhecimentos de Engenharia de Computação**

O projeto visa abordar um problema real e atual na automação industrial e em outras áreas que utilizam robótica, como na medicina e no ensino de robótica em sala de aula. A capacidade de programar um braço robótico para executar trajetórias pré-definidas pode melhorar a eficiência e a precisão de várias operações.

O desenvolvimento de um sistema de controle para um braço robótico é um desafio interdisciplinar que envolve eletrônica, programação, controle e mecânica, possibilitando assim aplicar os conhecimentos adquiridos ao longo do curso de maneira prática.

- **Exploração do Microcontrolador ATmega2560**

O ATmega2560 é um microcontrolador amplamente utilizado em projetos embarcados devido à sua versatilidade e recursos. Este trabalho oferece a oportunidade de explorar o uso da programação em linguagem C (a nível de registradores – *bare-metal*), bem como o uso de Modulação por Largura de Pulso (do inglês, *Pulse Width Modulation* – PWM) para controlar os servos do braço robótico.

- **Uso da Memória EEPROM**

A utilização da memória EEPROM para armazenar trajetórias é uma abordagem interessante que possibilita a reutilização e persistência de trajetórias pré-definidas. Isso pode ser útil em cenários onde várias tarefas distintas precisam ser executadas de forma automatizada.

- **Ferramenta prática de laboratório**

Uma sólida justificativa para inclusão deste braço robótico no escopo deste trabalho reside na sua potencial utilidade como uma ferramenta prática de laboratório. Atualmente, não dispomos de um recurso similar em nosso laboratório, o que limita significativamente nossas capacidades de pesquisa e ensino.

1.2 OBJETIVOS

- **Objetivo Geral:**

- O objetivo geral deste trabalho é desenvolver um sistema embarcado para controle manual e automático de um manipulador robótico.

- **Objetivos Específicos:**

- Programar o microcontrolador ATmega2560 em linguagem C, a nível de registradores, para ser capaz de receber e processar os comandos do usuário por meio dos potenciômetros para o controle do manipulador robótico;
- Elaborar o sistema para gravar e reproduzir as trajetórias dos servomotores de forma automática.

2 ESTADO DA ARTE

Neste capítulo são apresentados os tipos de manipuladores robóticos, essenciais para a escolha adequada do manipulador usado neste trabalho. Em seguida será abordado o microcontrolador ATmega2560, um componente chave para este projeto, compreendendo sua arquitetura e funcionalidades cruciais para projetos de robótica. Seguiremos com uma explanação sobre a memória EEPROM e sua importância na retenção de dados. Por fim, será dado uma breve explicação sobre o servo motor, componente essencial em muitos projetos de automação.

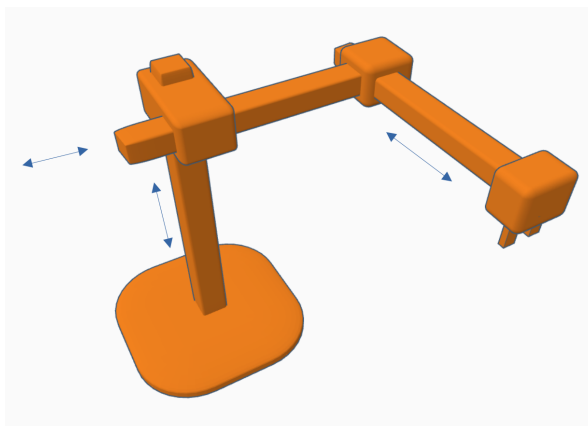
2.1 TIPOS DE MANIPULADORES

De acordo com Kato (2015), existem diferentes modelos conceituais de robôs manipuladores com características distintas em relação ao volume de trabalho, robustez, velocidade e exatidão nos movimentos. Os principais modelos são apresentados a seguir.

2.1.1 Cartesiano

Possui um volume de trabalho em forma de paralelepípedo e juntas prismáticas. É projetado para executar movimentos lineares ao longo de suas juntas prismáticas e possui 3 graus de liberdade. É preciso, mas relativamente lento. É resistente e adequado para tarefas com cargas pesadas (Kato, 2015).

Figura 1: Ilustração de um manipulador robótico do tipo cartesiano.

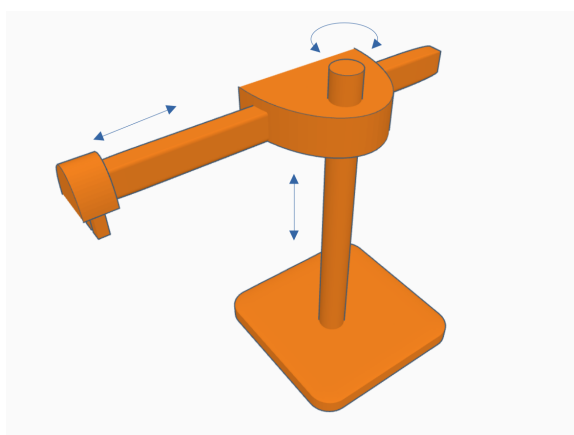


Fonte: Autor.

2.1.2 Cilíndrico

Tem um volume de trabalho próximo a um cilindro. Substitui a junta da base por uma junta rotativa. Oferece resistência mecânica e é adequado para carregar objetos não muito pesados. Pode ser usado com acionamentos pneumáticos para maior esforço (Kato, 2015).

Figura 2: Ilustração de um manipulador robótico do tipo cilíndrico.

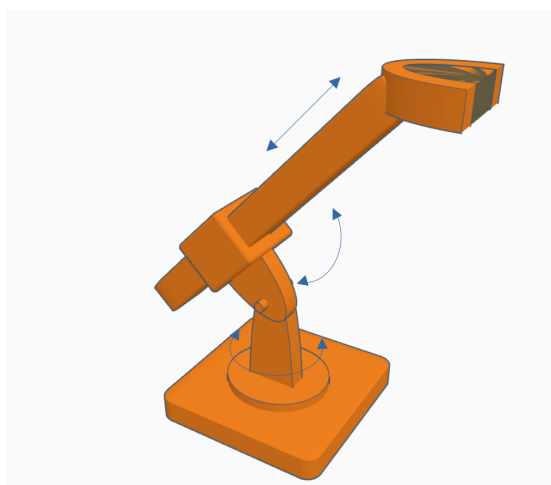


Fonte: Autor.

2.1.3 Esférico

Possui uma junta rotacional, proporcionando um volume de trabalho em forma de semi-esfera. Perde um pouco de resistência mecânica e exatidão, mas oferece movimentos rápidos e trajetórias simples (Kato, 2015).

Figura 3: Ilustração de um manipulador robótico do tipo esférico.

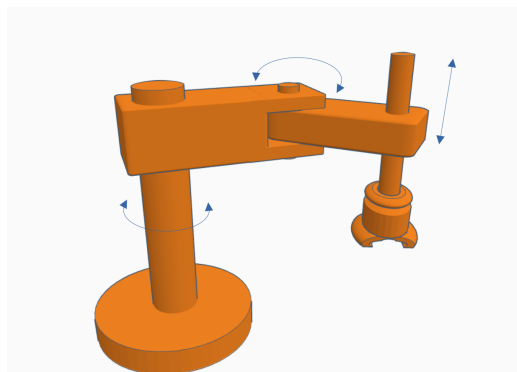


Fonte: Autor.

2.1.4 SCARA

Significa "*Selective Compliance Assembly Robot Arm*" (Braço de Robô para Montagem com Complacência Seletiva). Possui as mesmas juntas do modelo esférico, mas com eixos paralelos entre si. É resistente na direção vertical e é usado para mover objetos pequenos (Kato, 2015).

Figura 4: Ilustração de um manipulador robótico do tipo SCARA.

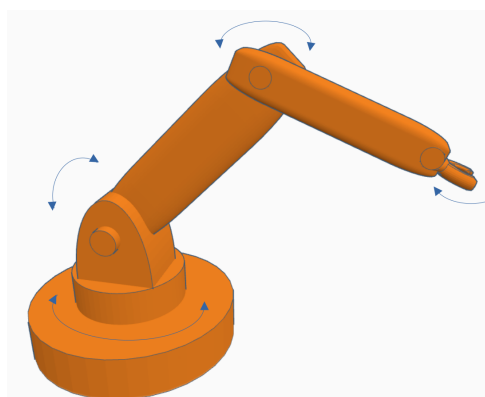


Fonte: Autor.

2.1.5 Antropomórfico

Assemelha-se ao braço humano, com juntas rotativas. A junta da base é ortogonal às demais. É ágil, mas a exatidão varia dentro do volume de trabalho (Kato, 2015). Este tipo de manipulador robótico foi utilizado no desenvolvimento deste trabalho, como será mostrado adiante.

Figura 5: Ilustração de um manipulador robótico do tipo antropomórfico.



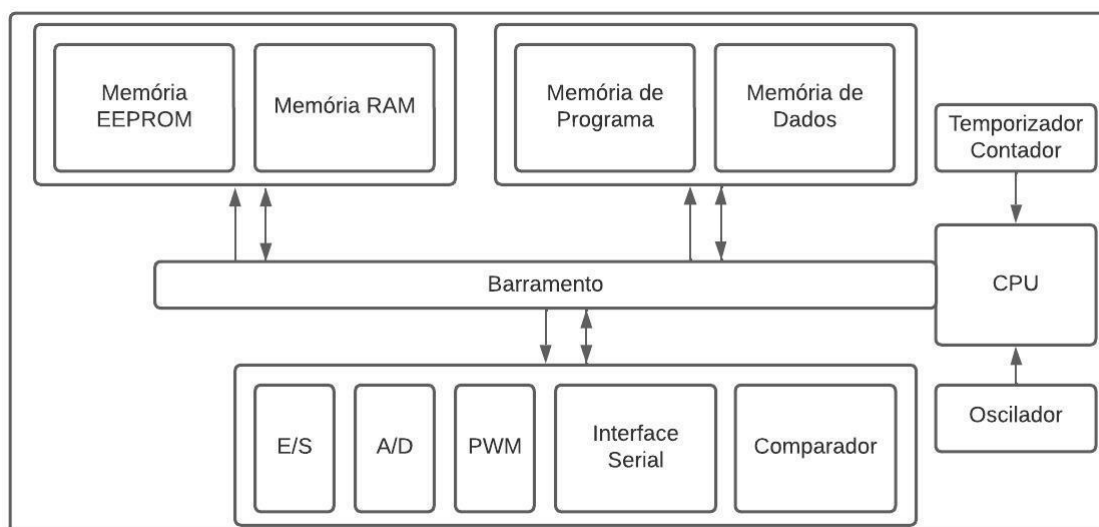
Fonte: Autor.

Esses modelos oferecem opções variadas para atender a diferentes necessidades de aplicação, levando em consideração fatores como exatidão, velocidade, resistência mecânica e tipo de movimento requerido.

2.2 O MICROCONTROLADOR ATmega2560

Um microcontrolador é um dispositivo microprocessado compacto que reúne diversas funcionalidades em um único chip, incluindo CPU, memória, temporizadores e circuitos de *clock*. Geralmente, requer apenas a adição de um cristal externo para determinar a frequência de operação. Sua principal vantagem é a capacidade de simplificar o desenvolvimento de sistemas eletrônicos, permitindo a criação rápida e eficiente de dispositivos com poucos componentes, devido à sua integração completa. Com o avanço tecnológico, os microcontroladores se tornaram amplamente usados em uma variedade de aplicações, desde produtos domésticos simples, como máquinas de lavar e televisores, até sistemas automotivos, eletrônicos portáteis como celulares e até mesmo em ambientes industriais, controlando máquinas e sistemas complexos (Lima e Villaça, 2012).

Figura 6: Diagrama básico da arquitetura de um microcontrolador.



Fonte: Autor.

Uma das famílias de microcontroladores muito utilizados ultimamente são os microcontroladores AVR. Os microcontroladores AVR surgiram em 1992 na Noruega, criados pelos estudantes de doutorado Alf-Egil Boden e Vegard Wollan. Eles refinaram o projeto por dois anos, depois foram para os EUA e a Atmel (agora parte da *Microchip Technology*) comprou a ideia. O primeiro AVR comercial, o AT90S1200, foi lançado em 1997. Os AVR são eficientes, com um núcleo compacto de 8 bits, ótimo desempenho, arquitetura RISC

(*Reduced Instruction Set Computer* – Computador com Conjunto de Instruções Reduzido) avançada, programação C amigável e muitos periféricos para diversas aplicações (Lima e Villaça, 2012). Os microcontroladores AVR ganharam destaque devido à sua ampla gama de recursos, tornando-os ideais para diversas aplicações, desde dispositivos portáteis até sistemas complexos de controle.

O ATmega2560 é um membro proeminente da família AVR, apresentando um conjunto expandido de recursos em comparação com seus predecessores. Ele é amplamente escolhido em projetos que demandam maior poder computacional, capacidade de E/S (entrada/saída) e memória. Abaixo estão alguns aspectos-chave que fazem do ATmega2560 uma escolha atraente para projetos de engenharia, como o sistema embarcado para o controle de braço robótico desenvolvido neste trabalho.

- A arquitetura RISC simplifica a execução de instruções ao utilizar várias linhas de código para completar a mesma tarefa em comparação com os CISC. Isso resulta em maior velocidade, já que os processadores RISC normalmente executam uma instrução por ciclo de *clock*, ao contrário dos CISC (*Complex Instruction Set Computer* – Computador com conjunto de Instruções Complexas), que demandam múltiplos ciclos de *clock* para uma única instrução (Lima e Villaça, 2012). Isso contribui para a rápida execução de códigos e, conseqüentemente, para a otimização do controle de sistemas em tempo real.
- O ATmega2560 possui uma grande quantidade de pinos de E/S, permitindo a conexão de diversos componentes externos, como sensores, atuadores e periféricos, necessários para o controle e a interação com o braço robótico.
- O ATmega2560 oferece vários contadores com recursos de modulação por largura de pulso (PWM). Isso é vital para controlar a velocidade e a posição dos servos do braço robótico, permitindo movimentos suaves e precisos.
- Com uma ampla memória Flash para armazenar o código do programa e memória EEPROM para armazenamento de dados persistentes, como trajetórias de movimento, o ATmega2560 suporta a implementação de projetos complexos e versáteis.
- Os temporizadores internos e as interrupções programáveis permitem a execução precisa de tarefas em momentos específicos, como atualizações de controle e captura de eventos.

2.2.1 EEPROM

As memórias EEPROM são uma evolução das primeiras versões de memória, conhecidas como ROM (*Read Only Memory* – Memória Apenas de Leitura), que só podiam ser programadas durante a fabricação. Logo depois, veio a EPROM (*Erasable Programmable Read-Only Memory* – Memória somente de leitura programável e apagável), que possibilita múltiplas gravações com apagamento por luz UV. Por fim, surgiu a EEPROM (*Electrically-Erasable Programmable Read-Only Memory* – Memória somente de leitura programável eletricamente e apagável), que pode ser apagada e programada eletricamente (Perim e Nascimento, 2017).

A memória EEPROM em um microcontrolador é usada para armazenar dados que precisam ser acessados pelo programa e não podem ser perdidos. Isso é útil para armazenar senhas, valores de sensores e outros dados que precisam ser preservados. No entanto, a EEPROM é lenta em termos de escrita, levando cerca de 3,3 ms para gravar um dado, o que pode ser demorado em algumas aplicações. Isso pode ser contornado com uso da interrupção da EEPROM para notificar quando uma escrita foi concluída (Lima e Villaça, 2012). Isso permite que o programa continue sem ter que monitorar ativamente o processo de gravação.

O ATmega2560 possui uma EEPROM com uma capacidade de 4 KB (4096 Bytes), permitindo armazenar uma quantidade razoável de dados. Porém, é importante ressaltar que há uma limitação de 10.000 escritas ou apagamentos de dados.

2.2.1 Modulação por Largura de Pulso (PWM)

A Modulação por Largura de Pulso (PWM) é uma estratégia empregada para regular saídas de forma contínua ou gradual, mediante o uso exclusivo de um sinal digital. Essa técnica viabiliza o ajuste da velocidade de um motor ou da intensidade luminosa de um LED, por exemplo. O sinal PWM consiste em uma forma de onda quadrada com frequência constante, porém, seus intervalos de tempo em que se encontra em estado alto (positivo) e em estado baixo (negativo) variam, sendo complementares entre si (Perim e Nascimento, 2017).

O ATmega2560 possui vários pinos que podem ser configurados para gerar saídas PWM usando seus *timers* internos. No total são 6 *timers*, dois de 8-bits e 4 de 16-bits, sendo os *Timer/Counter* 0 e *Timer/Counter* 2 de 8-bits e os *Timer/Counter* 1, *Timer/Counter* 3, *Timer/Counter* 4 e *Timer/Counter* 5 de 16-bits. Cada um desses *timers* possuem modos de operação que afetam a resolução e a frequência do sinal PWM gerado.

No modo *Clear Timer on Compare* ou CTC, o registrador OCRnx (*Output Compare Register* – Registrador de comparação de saída) é empregado para ajustar a precisão do contador. Em modo CTC, o contador é resetado quando seu valor coincide com o valor em OCRnx (MICROCHIP, 2017). Dessa forma é possível ter um maior controle da frequência do sinal.

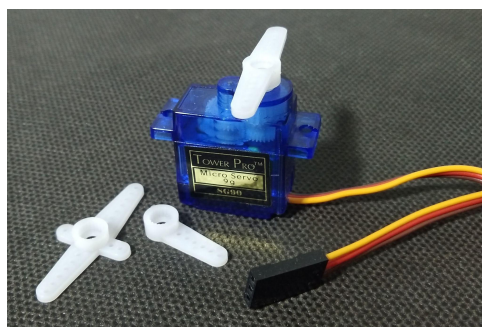
Outro modo é o PWM rápido (*Fast PWM*), configurado com WGM02:0 (*Waveform Generation mode* – Modo de geração de forma de onda) igual a 3 ou 7. oferece a capacidade de gerar um sinal PWM de alta frequência. Nesse modo, o PWM opera em uma única direção, contando de *BOTTOM* até *TOP* e, em seguida, reiniciando de *BOTTOM*. O valor de *TOP* é definido com 0xFFFF quando WGM2:0 = 3 e como o valor em OCRnx quando WGM2:0 = 7 (MICROCHIP, 2017). O PWM rápido é ideal para uma ampla variedade de aplicações, incluindo controle de velocidade de motores, controle de luminosidade de LEDs e muito mais.

Por fim, o modo de PWM com correção de fase fornece uma maneira de criar um PWM de alta precisão com correção de fase. Nesse modo, o contador opera em duas direções, indo de *BOTTOM* para *TOP* e depois de *TOP* para *BOTTOM*, com o valor de *TOP* sendo definido para 0xFFFF ou OCRnx, dependendo das condições (MICROCHIP, 2017).

2.4 SERVO MOTOR

Servos motores são dispositivos que oferecem movimento preciso a um eixo, sendo ideais para posicionamento preciso de peças mecânicas. Eles são compostos por um motor de corrente contínua com ímãs permanentes, um circuito de controle interno completo com drivers e engrenagens de redução (Lima e Villaça, 2012). Sua versatilidade e capacidade de posicionamento os tornam componentes indispensáveis em uma ampla gama de aplicações tanto no mundo da eletrônica quanto na automação.

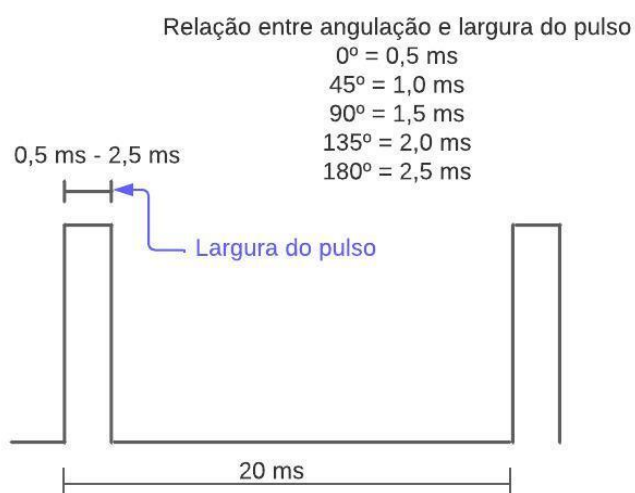
Figura 8: Servo motor.



Fonte: Autor.

Geralmente os servos motores têm três fio que desempenham funções específicas. Dois fios são utilizados para a alimentação do motor (Vcc e GND), onde a alimentação de operação é na faixa de 4,8 e 6 V, de acordo com o *datasheet*. O terceiro fio é usado para enviar comandos de posição ao servo, que se trata de pulsos PWM de um controlador externo (microcontrolador, por exemplo). A duração dos pulsos PWM determina a posição angular desejada para o servo motor, como ilustrado na Figura 9. É importante ressaltar que para o correto funcionamento do servo motor é necessário que o PWM opera na frequência certa, 50Hz geralmente. Ou seja, um comprimento de onda de 20 ms.

Figura 9: Sinal PWM para o controle do servo motor.



Fonte: Autor.

3 TRABALHOS RELACIONADOS

O processo construtivo, o desenvolvimento do sistema de um braço robótico e seu controle são áreas que visam proporcionar maior flexibilidade, mobilidade e interatividade para os usuários desses dispositivos. O desenvolvimento do braço robótico envolve a criação e montagem de seus componentes, como estruturas mecânicas, juntas, atuadores e sensores. O controle do braço robótico envolve a programação de algoritmos e a implementação de sistemas que permitam comandar os movimentos do braço de forma precisa e eficiente.

A literatura sobre o assunto apresenta diversos trabalhos que exploram esses conceitos, utilizando diferentes componentes e plataformas. Por exemplo, Melo et al. (2019) descrevem o desenvolvimento de um braço robótico aplicado na área da robótica industrial, composto por um protótipo mecânico e computacional programado por um FPGA que comanda o movimento de cada servo motor. O sistema é controlado por um programa em Java que utiliza uma conexão serial e um ambiente amigável ao usuário.

Bajerski e Abella (2010) propõem a construção de um pequeno braço robótico didático com uma interface de manipulação sem fio e inovadora, utilizando o controle remoto do console Nintendo Wii como dispositivo de manipulação remota do braço robótico. O objetivo é aplicar na prática conceitos de automação com materiais de fácil acesso e baixo custo. O braço robótico possui cinco graus de liberdade e é controlado por uma placa didática baseada em um microcontrolador MSP 430F2619. O braço robótico utiliza servomotores de posição para a movimentação das juntas e uma estrutura de alumínio para o suporte dos motores.

Oliveira (2014) construiu um braço robótico didático com três graus de liberdade controlado por Arduino, utilizando peças e ferramentas adequadas. O autor utilizou servomotores para o acionamento das juntas, uma garra com dois dedos para a manipulação de objetos e um joystick para o controle manual do braço. O autor também implementou um planejamento de trajetória automático para o braço robótico.

Esses trabalhos mostram que há uma variedade de possibilidades para o controle e o desenvolvimento de braços robóticos, utilizando diferentes tecnologias e plataformas. Essas possibilidades podem trazer benefícios para as indústrias e para os usuários finais, como maior precisão, eficiência e facilidade de uso. No entanto, também há desafios e limitações envolvidos nesse tipo de controle, como a necessidade de garantir a confiabilidade, a

estabilidade e a segurança da comunicação, bem como a compatibilidade entre os diferentes dispositivos e sistemas.

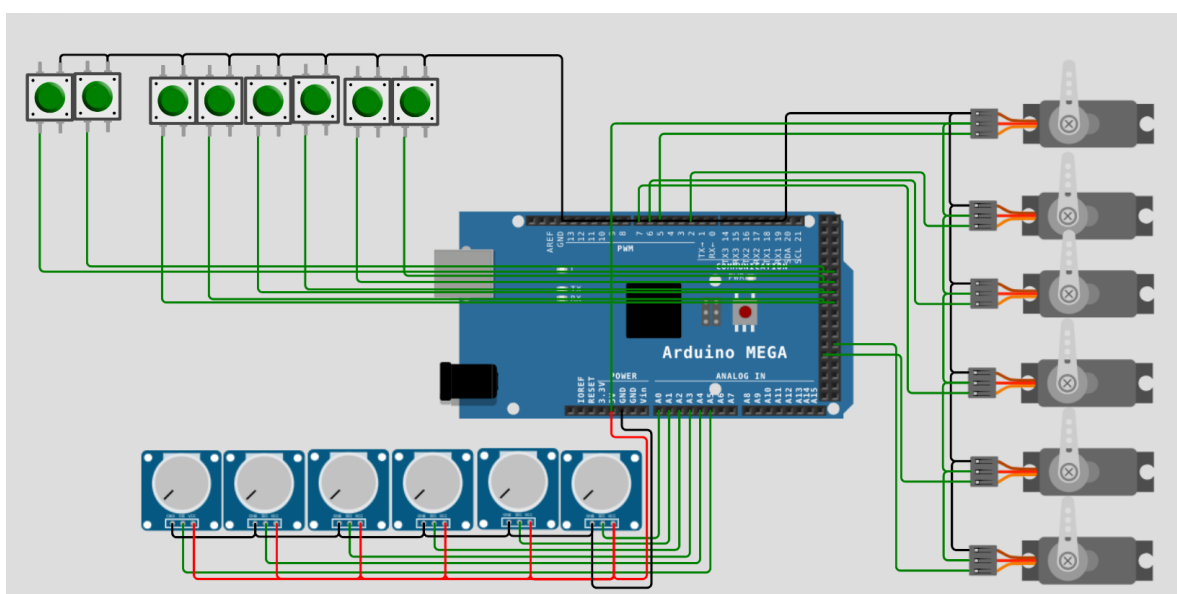
4 PROJETO DO MANIPULADOR ROBÓTICO

O sistema desenvolvido neste trabalho operou em dois modos principais: modo manual, permitindo o controle manual dos servos e a gravação das posições dos mesmos na EEPROM, e modo de execução automática, onde as trajetórias são lidas da EEPROM e os movimentos dos servos são realizados automaticamente.

Adiante será detalhado a arquitetura geral do sistema, abordando sua estrutura e componentes-chave. Além disso, será explorado a geração do sinal PWM, um elemento essencial para o controle preciso dos servomotores, e será discutido o desenvolvimento do firmware, descrevendo as decisões de implementação ao criar o código responsável por gerenciar o funcionamento do sistema.

4.1 ARQUITETURA GERAL DO SISTEMA

Figura 10: Arquitetura do sistema.



Fonte: Autor.

A arquitetura do sistema ilustrada na Figura 5 é composta pelos seguintes componentes:

- **ATmega2560:** O Arduino Mega foi usado apenas como placa programadora para carregar o código compilado no ATmega2560.
- **Servos Motores (6x):** Os servos foram os atuadores responsáveis pela movimentação do manipulador robótico de 6 graus de liberdade. Eles são controlados por saídas PWM dos pinos do ATmega2560.
- **Potenciômetros de 15 K Ω (6x):** Foram usados como entradas analógicas para permitir o controle manual dos servos motores. Cada potenciômetro corresponde a um dos 6 graus de liberdade.
- **Botões (8x):** Dois botões foram usados para alternar entre os modos manual e automático, e os seis demais botões foram usados para informar qual servo terá sua posição gravada na EEPROM.

4.2 GERAÇÃO DE SINAL PWM

A geração de sinal PWM por meio do hardware do microcontrolador ATmega2560 desempenhou um papel fundamental no controle preciso dos servomotores no sistema desenvolvido. Essa técnica de modulação permitiu a variação da largura de pulso dos sinais de controle, determinando assim a posição angular dos servos. Este processo foi essencial para garantir a precisão dos movimentos do manipulador robótico.

O ATmega2560 possui vários *timers*/contadores de 16 bits, que desempenham um papel crucial na geração de sinais PWM. Cada *timer*/contador foi configurado para operar em modo de comparação, permitindo assim a criação de um sinal PWM com alta resolução.

Antes de configurar a geração de PWM, foi importante escolher o modo de operação desejado. Como se trata de um sinal PWM para o controle de servomotores, é comum o uso do *Fast PWM*.

4.2.1 Configuração do valor de *Prescaler*

O próximo passo é definir o valor do *prescaler*, que determina a taxa na qual o contador será incrementado. De acordo com o *datasheet* do ATmega2560, é necessário configurar os *bits* CSn2, CSn1 e CSn0 dos registradores de controle dos *Timer/Counters* (TCCRnx - *Timer/Counter Control Register*) usados para definir o valor do *prescaler* que será usado, mostrado na Tabela 1.

CSn2	CSn1	CSn0	Descrição
0	0	0	Nenhuma fonte de <i>clock</i> . (Temporizador/contador parado)
0	0	1	$\text{clk}_{\text{IO}}/1$ (Sem <i>prescaling</i>)
0	1	0	$\text{clk}_{\text{IO}}/8$ (Do <i>prescaler</i>)
0	1	1	$\text{clk}_{\text{IO}}/64$ (Do <i>prescaler</i>)
1	0	0	$\text{clk}_{\text{IO}}/256$ (Do <i>prescaler</i>)
1	0	1	$\text{clk}_{\text{IO}}/1024$ (Do <i>prescaler</i>)
1	1	0	Fonte de <i>clock</i> externa no pino Tn. <i>Clock</i> na borda de descida
1	1	1	Fonte de <i>clock</i> externa no pino Tn. <i>Clock</i> na borda de subida

Tabela 1: Descrição do bit de seleção do *clock*.

Para este trabalho foi utilizado um *prescaler* de 8, o que significa que a frequência do *clock* foi dividida por 8. Dessa forma, considerando um cristal oscilador externo de 16 MHz, o contador do *timer* é incrementando a uma velocidade de 2 MHz (16 MHz/8), ou seja, uma velocidade de incremento mais rápida em relação aos outros valores de *prescaler*.

4.2.2 Configuração dos *Timers/Contadores*

Após a definição do valor do *prescaler*, os bits WGM0, WGM1 e WGM2 dos registradores de controle do *Timer/Counter* foram configurados para operar no modo PWM

rápido, com o valor máximo de contagem definido como OCRA (modo 7), seguindo as especificações detalhadas na Tabela 2.

Modo	WGM2	WGM1	WGM0	Modo de operação do Timer/Counter	TOP	OCR _x atualiza em	Flag TOP ativado
0	0	0	0	Normal	0xFF	Imediato	MAX
1	0	0	1	PWM, correção de fase	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Imediato	MAX
3	0	1	1	PWM rápido	0xFF	BOTTOM	MAX
4	1	0	0	Reservado	–	–	–
5	1	0	1	PWM, correção de fase	OCRA	TOP	BOTTOM
6	1	1	0	Reservado	–	–	–
7	1	1	1	PWM rápido	OCRA	BOTTOM	TOP

Tabela 2: Descrição do bit do modo de geração de forma de onda.

Para gerar o sinal PWM, foram empregados os registradores TCCR3A e TCCR3B do Timer/Counter 3, bem como os registradores TCCR4A e TCCR4B do Timer/Counter 4, e ainda os registradores TCCR5A e TCCR5B do Timer/Counter 5. Conforme mencionado no capítulo 2.2.1, os servomotores tipicamente funcionam com um sinal PWM de 50 Hz. A frequência do sinal PWM pode ser calculada utilizando a seguinte equação:

$$f_{OCnxPWM} = \frac{f_{clk_{I/O}}}{N(1+TOP)} \quad (1)$$

onde,

$f_{clk_{I/O}}$: representa Frequência do processador;

N : representa o divisor *prescaler* (1, 8, 64, 256, ou 1024);

TOP : representa o valor máximo da contagem.

Dado que já temos a informação de que a frequência desejada é de 50 Hz, precisamos determinar o valor máximo de contagem que permitirá alcançar essa frequência. Para isso, isolamos a variável TOP e substituímos os valores da frequência do processador (16 MHz, proveniente do cristal oscilador externo) e do prescaler (8).

$$TOP = \frac{f_{clk_I/O}}{N \cdot f_{OCnxPWM}} - 1 \quad (2)$$

$$TOP = \frac{16 \cdot 10^6 \text{ Hz}}{8 \cdot 50 \text{ Hz}} - 1 = 39999$$

Dessa forma, conseguimos criar um sinal PWM no modo rápido com uma frequência de 50 Hz. É crucial destacar que os três *timers* mencionados são de 16 *bits*, uma vez que para controlar os servomotores é imprescindível um sinal PWM com essa frequência. Isso não seria viável com *timers* de 8 *bits*, já que a frequência mínima que poderiam gerar é aproximadamente 61 Hz. Isso valida a escolha do ATmega2560 em vez de um microcontrolador mais básico, como o ATmega328p, uma vez que o último possui apenas um *timer* de 16 *bits*, enquanto o ATmega2560 dispõe de 4 *timers* de 16 *bits*. Dado que este trabalho requer no mínimo 3 *timers* de 16 bits, a opção pelo ATmega2560 se tornou essencial.

4.3 DESENVOLVIMENTO DO *FIRMWARE*

O *firmware* foi desenvolvido no sistema operacional Ubuntu, utilizando programação C no estilo *bare-metal*. Isso significa que o código foi escrito diretamente para o microcontrolador ATmega2560, sem depender de uma camada de abstração ou IDE (*Integrated Development Environment*). Essa abordagem permitiu um controle mais preciso do *hardware* e uma otimização mais eficiente do código.

Para compilar o código-fonte, foi utilizado o compilador *avr-gcc*, uma ferramenta amplamente recomendada para desenvolvimento em microcontroladores AVR. A utilização do *avr-gcc* permitiu a geração de código eficiente e otimizado para o ATmega2560. Após a

compilação, o código foi carregado no microcontrolador utilizando o *avrdude*, uma ferramenta de programação AVR.

O Arduino foi utilizado como placa programadora para gravar o código no microcontrolador. Embora tenha sido usado como uma ferramenta de programação, é importante observar que o código desenvolvido não dependeu da plataforma Arduino, sendo totalmente independente e escrito em C puro.

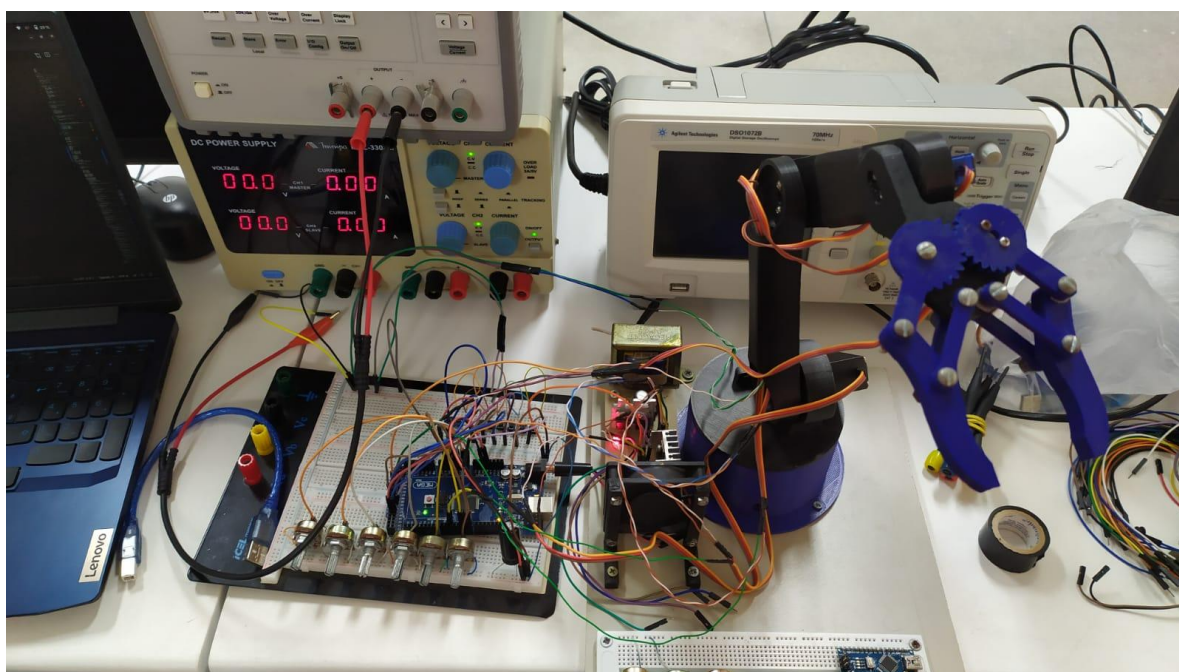
Uma característica importante deste trabalho é a geração de sinais PWM para controlar os servomotores e a conversão analógico-digital (A/D) para ler os valores dos potenciômetros. É relevante mencionar que nenhuma biblioteca externa foi utilizada para essas funcionalidades. O código inclui implementações personalizadas para gerar o sinal PWM necessário para controlar os servomotores e para realizar a conversão A/D dos valores dos potenciais.

Outra funcionalidade importante deste *firmware* é a gravação de dados na EEPROM interna do ATmega2560. Para essa tarefa, foi utilizada a biblioteca nativa “*avr/eeprom.h*”, que simplifica o acesso e a manipulação da EEPROM do microcontrolador. Isso permitiu o armazenamento de informações das posições das trajetórias de forma permanente.

5 RESULTADOS E DISCUSSÕES

O manipulador robótico utilizado neste trabalho foi confeccionado previamente, por meio da tecnologia de impressão 3D disponível na universidade. Vale destacar que o processo construtivo do manipulador não fez parte do escopo deste trabalho, uma vez que o foco estava no desenvolvimento e implantação do sistema embarcado. Na Figura 11 é apresentado o sistema geral para o controle do manipulador robótico montado em uma *protoboard*.

Figura 11: Visão geral do sistema.



Fonte: Autor.

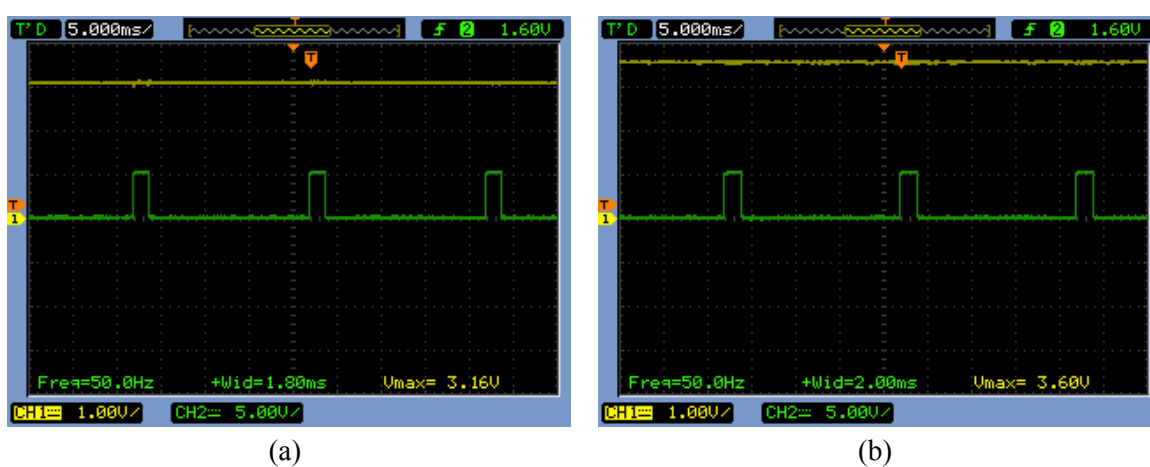
5.1 CONTROLE MANUAL

No que diz respeito ao controle manual do manipulador robótico por meio dos potenciômetros, os resultados obtidos foram promissores. Os operadores conseguiram manipular o robô de forma precisa e suave usando as interfaces de controle.

Os movimentos eram fluidos, e o sistema respondia de maneira rápida e eficiente aos comandos. Na Figura 13 é apresentado um conjunto de imagens do controle manual do manipulador robótico, demonstrando o movimento de todos os seis eixos de rotação.

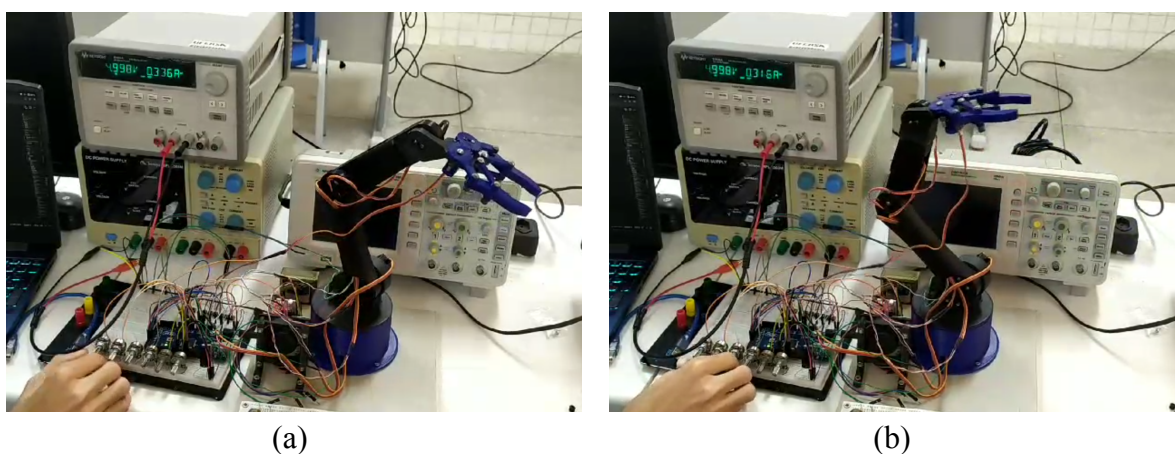
Os sinais PWM foram gerados com sucesso para cada um dos servomotores, na frequência de 50 Hz, sendo possível variar a largura do pulso entre 0,5 ms a 2,5 ms. A Figura 12 apresenta a leitura de um dos sinais PWM no osciloscópio operando em 50 Hz, onde é mostrado o sinal com dois valores diferentes de largura de pulso.

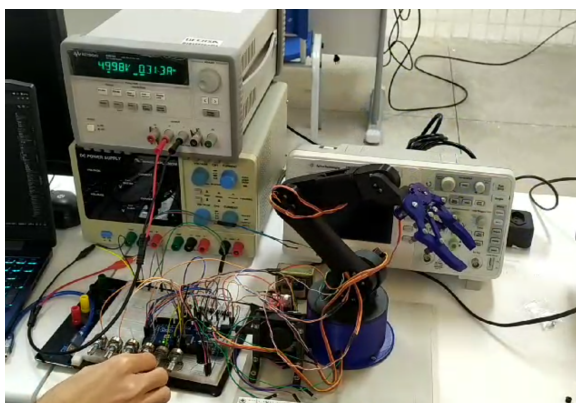
Figura 12: Leitura do sinal PWM gerado pelo ATmega2560



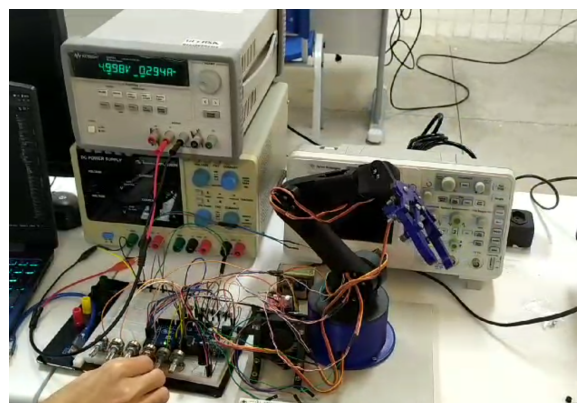
Fonte: Autor.

Figura 13: Controle manual do manipulador robótico.

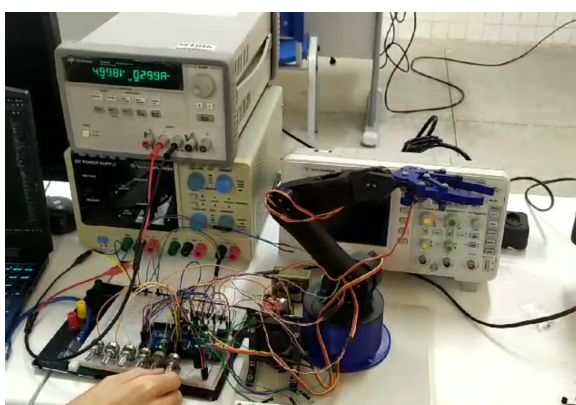




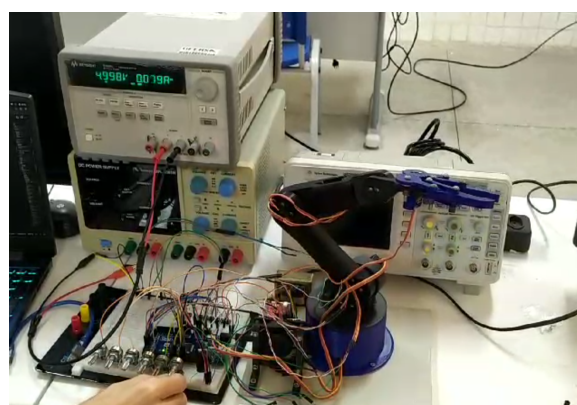
(c)



(d)



(e)



(f)

Fonte: Autor.

No entanto, durante essa fase de testes, observou-se uma questão de estabilidade. Ao tocar nos fios na *protoboard*, o sistema exibia pequenas oscilações nos movimentos do robô. Essa instabilidade foi identificada como um desafio a ser superado para melhorar o desempenho geral do sistema.

5.2 MODO AUTOMÁTICO

A implementação do modo automático, que envolvia a gravação e execução automática das posições dos servomotores, foi bem-sucedida. As trajetórias foram registradas de forma precisa na EEPROM do microcontrolador, e o sistema foi capaz de reproduzi-las com alta fidelidade.

Isso representou um avanço significativo em direção à automação do manipulador robótico, pois permitiu a execução de tarefas sem intervenção humana. No entanto, durante os testes, ficou evidente que a alimentação fornecida pelo microcontrolador não era suficiente para alimentar os servomotores de maneira ideal. Foi necessário recorrer a uma alimentação de uma fonte externa para garantir o desempenho adequado do sistema.

Um adendo importante a ser considerado na implementação bem-sucedida do modo automático de gravação e execução das posições dos servomotores é a limitação de escritas da memória EEPROM do microcontrolador. Uma vez que as trajetórias foram registradas com alta precisão na EEPROM, é fundamental lembrar que essa memória tem um número limitado de escritas até atingir o fim de sua vida útil. Portanto, ao utilizar a função *"eeprom_update_block"* para gravar as trajetórias, é importante para garantir uma vida útil adequada da memória EEPROM, evitando reescrever o mesmo dado em uma posição da memória.

Uma alternativa eficaz para contornar a limitação da EEPROM e estender a capacidade de armazenamento de trajetórias seria utilizar uma memória externa, como um cartão de memória flash. Isso permitiria uma maior flexibilidade e expansão do espaço de armazenamento, uma vez que os cartões de memória flash são conhecidos por sua capacidade substancial e durabilidade. O sistema poderia ser projetado para gravar as trajetórias diretamente no cartão de memória, evitando assim a preocupação com o limite de escritas da EEPROM interna e, ao mesmo tempo, permitindo uma capacidade de armazenamento muito maior para trajetórias complexas. Essa abordagem garantiria uma operação mais confiável e acomodaria a crescente necessidade de armazenamento de dados no sistema sem comprometer sua vida útil.

Apesar dos desafios iniciais, o sistema conseguiu cumprir os objetivos propostos, proporcionando um controle manual suave e preciso, juntamente com a automação das trajetórias. Além disso, as limitações que foram identificadas, como a instabilidade e a necessidade de alimentação externa, podem ser tratadas de forma eficaz. Isso se deve ao fato de que este trabalho apresenta um protótipo para um sistema para controle manual e automático de um manipulador robótico. No caso da instabilidade, decorrente do uso de *protoboard*, essa questão pode ser prontamente solucionada por meio da utilização de uma PCB adequada.

6 CONSIDERAÇÕES FINAIS

Neste trabalho, buscou-se projetar e desenvolver um sistema embarcado para controle manual e automático para um manipulador robótico, permitindo controle manual via potenciômetros e gravação/execução automática das posições dos servomotores usando a EEPROM presente no microcontrolador. O microcontrolador ATmega2560 foi crucial, programado em linguagem C a nível de registradores, para processar comandos e gerar sinais PWM com precisão.

Durante o controle manual, os potenciômetros possibilitaram uma manipulação suave e precisa, porém, enfrentamos instabilidade relacionada ao uso de *protoboards*, uma questão passível de solução com a adoção de PCB adequada. No modo automático, conseguimos gravar e reproduzir com sucesso trajetórias dos servomotores, avançando na automação do manipulador.

Apesar dos desafios iniciais, os objetivos foram alcançados, proporcionando controle manual e automático com movimentos precisos. As limitações, como a instabilidade e a necessidade de alimentação externa para os servomotores, foram eficazmente abordadas. Esse trabalho contribui para a robótica, destacando a importância do controle preciso de trajetórias e demonstrando a utilidade das técnicas utilizadas.

A abordagem bare-metal possibilitou controle preciso do hardware e otimização eficaz do código, validando projetos personalizados de microcontroladores. Este trabalho abre portas para futuras pesquisas e aplicações práticas na emocionante área da robótica.

Embora tenhamos alcançado os objetivos propostos neste trabalho, há várias oportunidades de melhoria e direções para futuras pesquisas. A questão da estabilidade mecânica, especialmente relacionada ao uso de *protoboards*, ainda pode ser aprimorada. Adicionar sensores ao sistema pode torná-lo mais versátil e capaz de responder a diferentes condições e ambientes, possibilitando a aplicação de conceitos de sistemas de controle para melhorar a detecção e *feedback* durante o controle.

REFERÊNCIAS

- BAYER, F. M.; ECKHARDT, M.; MACHADO, R. *Automação de sistemas*. Santa Maria: UFSM, 2011. Disponível em:
 <<https://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/18451/material/Apostila%202.pdf>>. Acesso em: 27 jun, 2023.
- BAJERSKI, Igor. ABELLA, Vinicius Dal Bó. Braço robótico com controle remoto bluetooth. 2010. Disponível em:
 <<https://revistaseletronicas.pucrs.br/index.php/graduacao/article/view/7905/5590>>. Acesso em: 27 jun, 2023.
- COCOTA, J.A.N.J. Elementos de Robótica (Apostila): Engenharia de Controle e Automação. Universidade Federal de Ouro Preto. Disponível em:
 <<http://professor.ufop.br/sites/default/files/cocota/files/elemroboticaintro.pdf>>. Acesso em: 4 jul. 2023.
- FIA. Impressão 3D: O que é, como funciona e exemplos de aplicações. 2021. Disponível em:
 <<https://fia.com.br/blog/impressao-3d/>>. Acesso em: 6 jul. 2023.
- KATO, Lucas Kenzo. Projeto de um braço robótico para fins didáticos. Florianópolis: UFSC, 2015. Disponível em: <<https://core.ac.uk/download/pdf/78552479.pdf>>. Acesso em: 13 jul. 2023.
- LIMA, Charles Borges de; VILLAÇA, Marco Valério Miorim. AVR e Arduino: técnicas de projeto. 2. ed. Florianópolis: Ed. dos autores, 2012.
- MELO, R.G.C., BRITO, F.T., JÚNIOR, O.A.L. Desenvolvimento de um braço robótico para fins educacionais. 2019. Disponível em:
 <https://prpi.ifce.edu.br/nl/_lib/file/doc1488-Trabalho/DESENVOLVIMENTO%20DE%20UM%20BRACO%20ROBOTICO%20PARA%20FINS%20EDUCACIONAIS.pdf>. Acesso em: 27 jun, 2023.
- MICROCHIP. *Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V*. 2 de mar. de 2017. Disponível em:

<https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf>. Acesso em: 23 jun. 2023.

OLIVEIRA, Eduardo Lino de. Construção de um braço robótico didático com três graus de liberdade controlado por arduino. 2014. Disponível em:

<https://repositorio.utfpr.edu.br/jspui/bitstream/1/27312/1/CP_COAUT_2014_2_04.pdf>.

Acesso em: 27 jun. 2023.

OTTONI, A.L.C. Material de estudo: introdução à robótica. Disponível em:

<https://www.ufsj.edu.br/portal2-repositorio/File/orcv/materialdeestudo_introducaoarobotica.pdf>. Acesso em: 4 jul. 2023.

PERIM, V. G. de C.; NASCIMENTO, J. N. R. do. *Microcontroladores e Microprocessadores*. Londrina: Editora e Distribuidora Educacional S.A., 2017.

SANTOS, V. M. F. *Robótica industrial: apontamentos teóricos, exercícios para aulas práticas, problemas de exame resolvidos*. Aveiro: Universidade de Aveiro, 2003. Disponível em:

<<http://www.ece.ufrrs.br/%7Erventura/RoboticaIndustrial.pdf>>. Acesso em: 5 jul. 2023.

SIMPLÍCIO, P. V. G.; LIMA, B. R. Manipuladores robóticos industriais. Caderno de Graduação - Ciências Exatas e Tecnológicas - UNIT - SERGIPE, v. 3, n. 3, p. 85-92, 2016.

Disponível em: <<https://periodicos.set.edu.br/cadernoexatas/article/view/3572>>. Acesso em: 4 jul. 2023.