Desenvolvimento de um Braço Robótico controlado por Joystick

Dhomini Andrade dos Santos[1], Helena Patrícia de S. Silva[2], Daniel dos Anjos Costa[3]

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

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

[3] Colegiado de Informática – Instituto Federal da Bahia (IFBA) – Euclides da Cunha – BA – Brasil

Dhominikof@gmail.com, helena121280@gmail.com, daniel.anjos@ifba.edu.br

Abstract. *This work presents the implementation of a robotic arm with two different control mechanisms, designed to operate simultaneously. The following control mechanisms were developed: a local control using buttons on the board and a remote control with a PS2 joystick. To process the input signals from these mechanisms, we use the Arduino Uno. However, due to the limitation of digital inputs on the processing board, we used the 74HC148 coding integrated circuit as an intermediary to reduce the amount of local control signals directed to the Arduino Uno.*

Resumo. *Este trabalho apresenta a implementação de um braço robótico com dois diferentes mecanismos de controle, projetados para operar de forma simultânea. Foram desenvolvidos os seguintes mecanismos de controle: um controle local utilizando botões na placa e um controle remoto com um joystick de PS2. Para processar os sinais de entrada provenientes desses mecanismos, utilizamos o Arduino Uno. No entanto, devido à limitação de entradas digitais na placa de processamento, empregamos o circuito integrado de codificação 74HC148 como intermediário para reduzir a quantidade de sinais do controle local direcionados ao Arduino Uno.*

1. INTRODUÇÃO

No último século a ciência revolucionou com a invenção de diversas máquinas que ajudam o homem em várias tarefas, em especial as máquinas eletrônicas ou programáveis como é o caso dos computadores. Hoje em dia temos máquinas com as mais variadas funções que podem ser controladas por humanos ou inteligência artificial. No início da computação usava-se cartões perfurados na interação com o computador, atualmente temos disponíveis diversos dispositivos de entrada/saída como mouse, teclado, telas touchscreen (sensível ao toque), controle remoto, comandos por voz e até holograma interativo.

Na tentativa de aperfeiçoar cada vez mais a interação homem-máquina a indústria da informática está cada vez mais evoluindo os dispositivos de entrada e saída. Um dispositivo muito versátil para controle de máquinas é o joystick ou popularmente conhecido como controle de videogame. De acordo com Kim et al. (2020), essa alternativa tem sido estudada para aplicações como o controle de equipamentos médicos

e industriais. Por definição, ele é um dispositivo de entrada, utilizado em jogos de computador ou vídeo, dotado de uma alavanca capaz de controlar o movimento de um cursor na tela, e de um ou mais botões capazes de comandar certas ações, ao serem pressionados (DICIONÁRIO OXFORD, 2023).

Neste trabalho desenvolvemos um braço robótico controlado por Arduino e um Joystick na interação/controle entre homem e máquina. Exploramos diversas funcionalidades do joystick para melhor interação com o protótipo desenvolvido. Projetos como o nosso podem ser expandidos para uso da indústria no controle de ferramentas robóticas, pois o joystick é um ferramenta flexível e barata.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresentamos a Fundamentação Teórica que serve de base para o desenvolvimento do sistema proposto neste trabalho.

2.1. Arduino

De acordo com Nussey (2019), o arduino é uma plataforma de prototipagem eletrônica de código aberto que permite criar projetos interativos usando componentes eletrônicos, sensores e atuadores. Nesta seção descreveremos o Arduino UNO, pois foi o que usamos no projeto. Sua escolha foi influenciada por diversos fatores, incluindo a facilidade de uso, a disponibilidade abundante de bibliotecas, a acessibilidade e o preço acessível do dispositivo.

O Arduino UNO possui o Microcontrolador ATmega328P que é um componente de processamento responsável pela execução do código embarcado, interpretação de sinais recebidos e emissão de sinais atuadores (NUSSEY, 2019). As portas digitais são as portas de entrada/saída que operam com valores binários que podem ser “0” ou “1”. Essas portas são chamadas de GPIO (General Purpose Input/Output) e podem ser utilizadas para conectar e controlar dispositivos digitais como LEDs, botões, relés, motores, entre outros. Elas se encontram na placa com a numeração de 0 a 13.

As portas analógicas do Arduino UNO são as portas de entrada que operam com valores contínuos que variam em uma faixa de valores. Essas portas são chamadas de ADC (Analog to Digital Converter) e permitem que o Arduino possa ler sinais de sensores ou dispositivos que geram sinais analógicos, como sensores de temperatura, potenciômetros, fotocélulas, entre outros (NUSSEY, 2019). Elas se encontram na placa com as siglas A0, A1, A2, A3, A4 e A5.

2.2. Codificador

O codificador é um dispositivo eletrônico utilizado para converter sinais de entrada em códigos digitais (em binário), este podendo ser interpretado por outros sistemas (DORF e SVOBODA, 2012). A sua aplicação foi com o intuito de receber as entradas de dados e diminuir a quantidade de saídas. Vale ressaltar, que cada entrada do codificador refere-se ao estado de uma entrada diferente, e, por causa da propriedade lógica, apenas

uma pode ser acionada por vez. O valor de saída será um código binário que representa qual entrada está sendo acionada (SOUZA e FERREIRA, 2023).O codificador utilizado neste projeto foi o 74HC148, sendo ele do tipo 8-to-3, pois ele converte 8 sinais de entrada (0 a 7) para três pinos de saída de dados binários (A0 a A2) (DATASHEET DO 74HC148, 1992).

2.3. Joystick de Playstation 2 (PS2)

Segundo Guimarães (2017), o controle de PlayStation 2 (PS2) é um dispositivo de entrada que se comunica com o console de jogos por meio de uma porta PS2. O controle utiliza um protocolo de comunicação que é baseado em uma combinação de sinais digitais e analógicos para transmitir informações sobre os botões pressionados. Adicionalmente, beneficia-se de uma extensa comunidade de usuários e documentação, simplificando sua aplicação em uma variedade de projetos. O dispositivo oferece múltiplas opções de entrada e uma interface de utilização descomplicada.

Na Figura 01 é mostrado o esquema dos pinos de conexão nos terminais de saída/entrada do controle. Pela Figura podemos observar que existem 9 pinos de conexão e cada um com sua respectiva função. O pino “1” transmite sinais de dados do controle para o Playstation. Esse sinal é uma transmissão serial de 8 bits sincronizado pelo sinal de clock no pino 7 (CONTROLE de PS2 e Arduino, 2015). Para a ligação desse pino do arduino deve-se usar um resistor de 1Ω à 110kΩ. Esta resistência é necessária porque o controle só pode conectar este pino à terra, então ela se faz necessária para garantir o nível lógico 1 (CONTROLE de PS2 e Arduino, 2015).

O pino “2” é o pino de comando enviado do Playstation para o controle. Esse sinal é serial de 8 bits sincronizado pelo pino 7 de clock. O pino “3” recebe o sinal de vibração emitido do Playstation para o controle. O terminal “4” e “5” são respectivamente o GND (0V) e o VCC (3.3V ou 5V). O pino “6” é um sinal enviado do Playstation para o controle na tentativa de chamar sua atenção. O sinal desse pino permanece em nível lógico baixo durante o uso do controle (CONTROLE de PS2 e Arduino, 2015).

Pelo pino “7” é transmitido o sinal de clock enviado do Playstation para o controle entre as frequências 250kHz à 500kHz. O pino “8” não transmite sinais. E por fim o pino “9”, que transmite sinal analógico do controle para o Playstation (CONTROLE de PS2 e Arduino, 2015).

Figura 01 - Esquema de ligação dos terminais de conexão do controle de PS2.

3. DESENVOLVIMENTO

Nessa seção apresentaremos com detalhes todo o projeto físico e lógico do braço robótico desenvolvido. O circuito elétrico e eletrônico do braço foi implementado para receber comandos de dois mecanismos digitais de controle em paralelo, ou seja, o mesmo circuito foi projetado para ser utilizado e controlado por dois mecanismos de controle diferentes. O controle por Joystick é acionado manualmente, porém fora da placa. O controle Local só pode ser manuseado na própria placa onde estão os circuitos eletrônicos do braço.

O duplo acionamento e controle é bastante usado em máquinas industriais. Muitas máquinas são controladas localmente pelo operador ou remotamente por um outro operador distante da mesma.

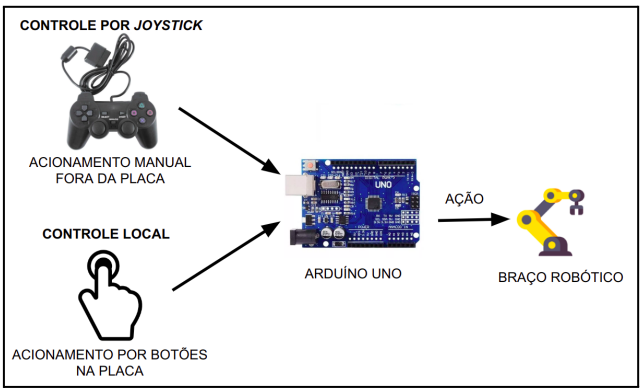

Figura 02 - Ilustração de alto nível da implementação dos circuitos de controle do braço robótico.

Para a movimentação das partes mecânicas do braço utilizamos quatro servomotores. Os servomotores são dispositivos de atuação de alta precisão que possibilitam o controle do seu ângulo de giro, o que o torna uma excelente ferramenta em aplicações de robótica.

O dispositivo de processamento utilizado foi o Arduino UNO. Para nosso projeto, a quantidade de portas digitais disponibilizadas pelo Arduino UNO é insuficiente, pois o projeto inicial demandaria 8 portas para botões do controle local, 5 portas para comunicação com o Joystick PS2 e 4 portas para controle dos servomotores, que daria 17 portas.

Para solucionar essa limitação de projeto, optamos em utilizar o codificador 74HC148 para reduzir a quantidade de portas necessárias nos sinais dos botões digitais. Um codificador é conhecido na eletrônica como o circuito integrado “tradutor” que converte determinadas sequências maiores de bits em sequências menores únicas.

A Figura 03 ilustra a organização esquemática dos módulos do sistema e dos sinais emitidos entre eles. Por ela, podemos observar (parte esquerda superior) que o Módulo dos Botões emite oito sinais elétricos digitais diferentes. Esses sinais são enviados por condutores para o segundo módulo, que é chamado de Módulo de

Codificação. Esse módulo tem a função de traduzir os sinais recebidos pelo módulo anterior em uma nova sequência de bits menor e única que será enviado para o Arduino UNO. Podemos observar pelo texto da figura que o módulo de codificação recebe oito bits de entrada e os converte em quatro bits de saída.

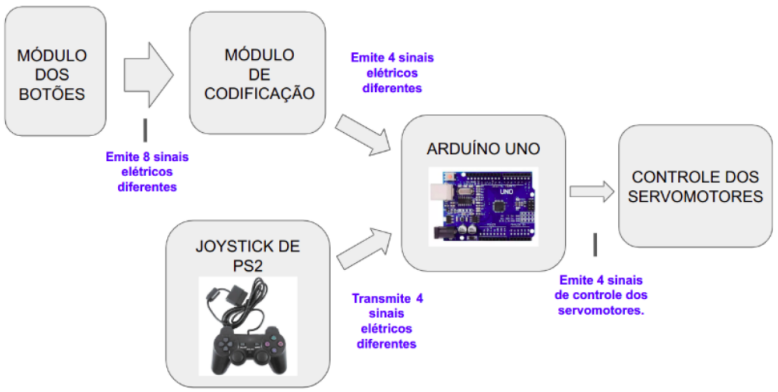

Figura 03 - Organização esquemática dos módulos do sistema.

Na parte esquerda inferior podemos observar o Joystick PS2 que demanda cinco canais de comunicação com o Arduino UNO. O Arduino UNO recebe os sinais do módulo de codificação e do Joystick, os interpreta e atua nos servomotores para movimentação do braço.

3.1. Implementação

Nesta seção apresentaremos em detalhes como foi implementado o circuito elétrico do sistema, os códigos embarcados no Arduino Uno e a implementação mecânica do braço robótico.

A montagem mecânica do braço robótico foi construída com as peças do modelo encontrado no blog Ultimaker Thingiverse (PHENOPTIX, 2014) que é de domínio público. Esse modelo foi escolhido por causa da sua ampla utilização em outros projetos disponíveis na literatura, o que reforçou para nossa equipe que as peças têm formatos e encaixes corretos para a montagem do braço. As peças do braço foram confeccionadas em MDF de 3mm.

É importante lembrar que o mesmo circuito físico foi utilizado com os dois tipos de controle implementados no projeto. A Figura 04 ilustra as interconexões dos diversos componentes utilizados na implementação do braço robótico, e a Tabela 02 associa os botões à sua codificação (emitida pelo módulo de codificação), motores e ações desencadeadas.

3.1.1. Controle Local

Iniciaremos nossa explicação pela montagem do Controle Local, pois ele foi implementado com botões (para interação do usuário) na mesma placa do circuito do braço. Quando um usuário aciona um dos botões, um sinal elétrico é emitido para o codificador que retorna uma combinação em BCD (de 3 dígitos mais o sinal SG) para o

Arduino UNO. O Arduino, por sua vez, interpreta as entradas e atua na movimentação de um dos quatro servomotores.

Por exemplo, observando a primeira linha da Tabela 02, se o usuário pressionar o botão B0 o módulo de codificação vai emitir o sinal de saída “111” que entrará no Arduino UNO. O código depositado dentro do Arduino UNO prevê que se ele receber essa entrada deve ser gerado no servomotor M0 a movimentação para a esquerda até um limite de ângulo pré-estabelecido. A mesma coisa acontece com se o usuário pressionar o botão B1, porém o Arduino controlará a movimentação do servomotor para a direita até um limite de ângulo pré-estabelecido.

Na Figura 04 podemos observar que as saídas dos botões B0, B1, B2, B3, B4, B5, B6 e B7 foram ligadas nas entradas 0, 1, 2, 3, 4, 5, 6 e 7 do codificador 74HC148 respectivamente. As saídas A0, A1, A2 e GS do codificador foram ligadas nas portas 2, 3, 4 e 5 do Arduino Uno. A porta Vcc do codificador foi alimentada com 5.0 volts, a porta GND com 0 volts e a entrada Ei recebeu 0 volts o que permite o acionamento do codificador.

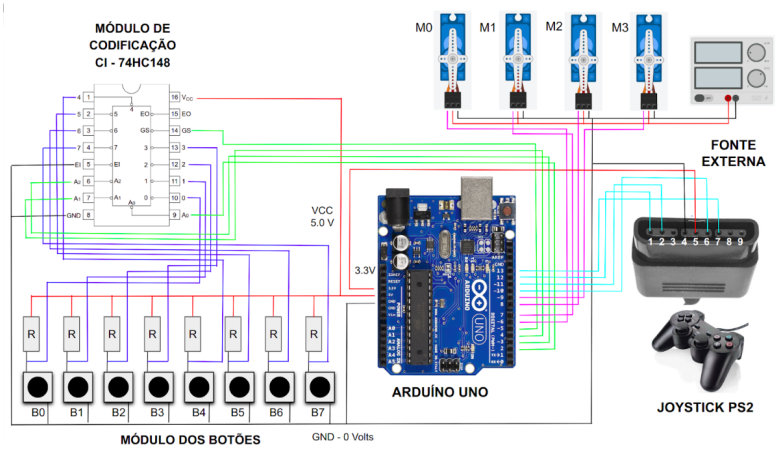

Figura 04 - Implementação do circuito eletrônico do braço robótico.

O Joystick PS2 possui um terminal de conexão que possui 9 pinos (explicados na seção 2.4. da Fundamentação Teórica). Os seus terminais 1, 2, 4, 5, 6 e 7 foram ligados respectivamente nos pinos 12, 11, GND com 0 volts, VCC com 3.3 volts, 10 e 13 do Arduino UNO. O Joystick recebe sua alimentação nos pinos 5 e 6 com uma diferença de potencial de 3.3V.

E finalmente os servomotores M0, M1, M2 e M3 foram ligados nos terminais 6, 7, 8 e 9 do Arduino, respectivamente. É importante também destacar que o projeto tem duas fontes externas e uma delas é exclusivamente para alimentar os servomotores. Por causa disso, foi necessário a interligação de todos os potenciais GND (0 volts).

Diante do circuito apresentado na Figura 04 e os dados disponibilizados na Tabela 02 detalharemos nessa seção o código do Controle Local que foi carregado no Arduino UNO apresentado na Figura 05.

Tabela 02 - Associa os botões do circuito aos seus respectivos motores e ações desencadeadas.

LISTA DE BOTÕES	CODIFICAÇÃO	SERVO DE ATUAÇÃO	AÇÃO
B0	111	M0	Movimentar o eixo do motor M0 para esquerda até um limite pré-estabelecido.
B1	110	M0	Movimentar o eixo do motor M0 para direita até um limite pré-estabelecido.
B2	101	M1	Movimentar o eixo do motor M1 para esquerda até um limite pré-estabelecido.
B3	100	M1	Movimentar o eixo do motor M1 para direita até um limite pré-estabelecido.
B4	011	M2	Movimentar o eixo do motor M2 para esquerda até um limite pré-estabelecido.
B5	010	M2	Movimentar o eixo do motor M2 para direita até um limite pré-estabelecido.
B6	001	M3	Movimentar o eixo do motor M3 para esquerda até um limite pré-estabelecido.
B7	000	M3	Movimentar o eixo do motor M3 para direita até um limite pré-estabelecido.

Na linha 4 criamos quatro objetos do tipo “Servo”, um para controlar cada servomotor do projeto, com os mesmos nomes utilizados na Figura 04 e Tabela 02, a saber: M0, M1, M2 e M3. Na linha 5 criamos quatro variáveis do tipo inteiro para armazenar os ângulos de cada servomotor após sua movimentação. Dentro da função “setup( )” atribuímos aos pinos 2, 3, 4 e 5 o status de pinos de entrada (INPUT) nas linhas 09 a 12. Esses pinos estão ligados às saídas A0, A1, A2 e GS do codificador 74HC148, conforme apresentado na Figura 04.

```
2  #include <Servo.h>
3
4  Servo M0, M1, M2, M3; // Criação dos objetos dos servomotores
5  int anguloM0, anguloM1, anguloM2, anguloM3; // Armazenam os ângulo dos motores
6
7  void setup()
8  {
9    pinMode(2, INPUT);  //pino de entrada que recebe o sinal A0 do Codificador
10   pinMode(3, INPUT);  //pino de entrada que recebe o sinal A1 do Codificador
11   pinMode(4, INPUT);  //pino de entrada que recebe o sinal A2 do Codificador
12   pinMode(5, INPUT);  //pino de entrada que recebe o sinal GS do Codificador
13
14   pinMode(6, OUTPUT); //pino de saída que vai atuar no M0
15   M0.attach(6, 500, 2500);
16   pinMode(7, OUTPUT); //pino de saída que vai atuar no M1
17   M1.attach(7, 500, 2500);
18   pinMode(8, OUTPUT); //pino de saída que vai atuar no M2
19   M2.attach(8, 500, 2500);
20   pinMode(9, OUTPUT); //pino de saída que vai atuar no M3
21   M3.attach(9, 500, 2500);
22 }
23
24 void loop()
25 {
26   controleLocal();
27   controleJoyStickPS2(); }
```

Figura 05 - Primeira parte do pseudocódigo do Controle Local.

Dentro da função “controleLocal() criamos 4 variáveis inteiras para armazenar o valor atual de leitura dos pinos 2, 3, 4 e 5 (Figura 06). Estes pinos estão ligados ao codificador e por isso nomeamos cada uma das variáveis com o mesmo identificador dos pinos do codificador, a saber: A0, A1, A2 e GS para facilitar o entendimento do leitor.

Na linha 36 da Figura 06 temos o primeiro condicional que verifica se o valor de GS é igual a zero. Conforme citado anteriormente, o sinal GS serve para indicar que a saída dos pinos A0, A1 e A2 são válidas. Portanto, quando ele tem valor lógico “0” as

saídas são válidas e quando ele tem valor lógico “1” as saídas são inválidas. O primeiro condicional (linha 36) garante que os motores só serão acionados por sinais válidos.

Das linhas 38 até 45 implementamos 8 condicionais em cascata para verificar a combinação dos códigos emitidos pelo codificador. Os códigos ilustrados na Figura 06 e as ações executadas dentro do condicional são as mesmas apresentadas na Tabela 02. Entretanto, após a movimentação dos servomotores é armazenado os seus ângulos nas variáveis “anguloM0”, “anguloM2”, “anguloM3” e “anguloM4” respectivamente.

Na linha 46 implementamos um “else” para executar possíveis entradas não previstas. Em seu corpo simplesmente desativamos a atuação dos servomotores.

```
29  void controleLocal()
30  {
31    int A0 = digitalRead(2);  //leitura do sinal digital A0 do Codificador
32    int A1 = digitalRead(3);  //leitura do sinal digital A1 do Codificador
33    int A2 = digitalRead(4);  //leitura do sinal digital A2 do Codificador
34    int GS = digitalRead(5);  //leitura do sinal digital GS do Codificador
35
36    if(GS == 0)
37    {
38      if(A2 == 1 && A1 == 1 && A0 == 1) { /*Movimenta o M0 para a ESQUERDA*/ }
39      else if(A2 == 1 && A1 == 1 && A0 == 0) { /*Movimenta o M0 para a DIREITA*/ }
40      else if(A2 == 1 && A1 == 0 && A0 == 1) { /*Movimenta o M1 para a ESQUERDA*/ }
41      else if(A2 == 1 && A1 == 0 && A0 == 0) { /*Movimenta o M1 para a DIREITA*/ }
42      else if(A2 == 0 && A1 == 1 && A0 == 1) { /*Movimenta o M2 para a ESQUERDA*/ }
43      else if(A2 == 0 && A1 == 1 && A0 == 0) { /*Movimenta o M2 para a DIREITA*/ }
44      else if(A2 == 0 && A1 == 0 && A0 == 1) { /*Movimenta o M3 para a ESQUERDA*/ }
45      else if(A2 == 0 && A1 == 0 && A0 == 1) { /*Movimenta o M3 para a DIREITA*/ }
46      else  {  /*Não faça nada*/  }
47    }
48  }
49
```

Figura 06 - Segunda parte do pseudocódigo do Controle Local.

3.1.2. Controle por Joystick de PS2

Nesta seção abordaremos a implementação do controle do braço robótico por Joystick de PS2. É importante destacar que não foram feitas alterações no circuito apresentado na Figura 04 para adicionar mais essa modalidade de controle, apenas foram acrescentadas uma biblioteca e novas funções no código embarcado no Arduino.

Na Figura 04 no canto direito podemos observar como foi a ligação física do controle de PS2 com o Arduino Uno. É importante lembrar ao leitor que todo o esquema dos pinos de conexão do controle de videogame PS2 são exibidos na Figura 01 da Fundamentação Teórica. Pela Figura 04 percebemos que os pinos 1, 2, 4, 5, 6 e 7 do terminal do joystick foram ligados nas portas 12, 11, GND (0V), VCC (3.3V), 10 e 13 do Arduino UNO respectivamente.

O código-fonte que exibe implementação da função “controleJoyStickPS2()” se encontra na Figura 07. Na linha 55 encontramos um condicional que recebe a condição “status == 0”, o que significa que o joystick está se comunicando sem erros. No corpo do condicional da linha 55 encontramos outros 8 condicionais em cascata que fazem a verificação de botões digitais que foram pressionados no joystick. Essas verificações são necessárias para que ocorra a movimentação nos servomotores do braço robótico.

O primeiro condicional (linha 57) verifica se o botão PSB_PINK (botão quadrado) foi pressionado. Se essa condição for verdadeira, o servomotor vai fazer seu

giro para sua esquerda (até um limite pré-estabelecido) pelo tempo que o botão estiver pressionado. Ao final, o ângulo atual do eixo do servomotor será armazenado na variável "anguloM0". O condicional da linha 58 executa o movimento contrário do servomotor M0, ou seja, para a direita enquanto a condição for verdadeira. O botão utilizado para a ativação do segundo condicional foi o PSB_RED (botão círculo). O mesmo padrão foi repetido nos demais condicionais das linhas 59 até 64, porém cada condicional associado a um botão digital do joystick.

```
53  void controleJoyStickPS2()
54  {
55    if(status == 0) // Controle encontrado e configurado
56    {
57      if(ps2x.Button(PSB_PINK)) { /* Movimente o M0 para a ESQUERDA */ }
58      else if(ps2x.Button(PSB_RED)) { /* Movimente o M0 para a DIREITA  */ }
59      else if(ps2x.Button(PSB_GREEN)) { /* Movimente o M1 para a ESQUERDA */ }
60      else if(ps2x.Button(PSB_BLUE)) { /* Movimente o M1 para a DIREITA */ }
61      else if(ps2x.Button(PSB_PAD_UP)) { /* Movimente o M2 para a ESQUERDA */ }
62      else if(ps2x.Button(PSB_PAD_DOWN)) { /* Movimente o M2 para a DIREITA */ }
63      else if(ps2x.Button(PSB_PAD_RIGHT)) { /* Movimente o M3 para a ESQUERDA */ }
64      else if(ps2x.Button(PSB_PAD_LEFT)) { /* Movimente o M3 para a DIREITA */ }
65    }
66  }
```

Figura 07 - Segunda parte do pseudocódigo do Controle Local.

O uso das variáveis globais "anguloM0", "anguloM1", "anguloM2" e "anguloM3" permite que os dois tipos de controle (por botões na placa e pelo joystick) funcionem em paralelo. Pois, elas armazenam o estado atual dos ângulos dos servomotores e são usadas nas funções "controleLocal()" e "controleJoyStickPS2()".

4. CONCLUSÃO

Neste projeto, foi realizada a implementação de um braço robótico com duas maneiras de controlá-lo: por meio de um controle de PS2 e um controle local, por meio de botões na protoboard. Para solucionar a limitação de entradas digitais do Arduino e implementar o controle local por botões na placa utilizamos o codificador 74HC148. Ele possibilitou reduzir de oito para quatro a quantidade de entradas digitais necessárias para implementação do controle local.

Utilizamos no projeto o Arduino Uno como componente de processamento de sinais de entrada e emissão de sinais de saída para os atuadores. É importante destacar que utilizamos neste projeto servomotores para atuar frente às partes mecânicas do braço. Eles são dispositivos atuadores que permitem controlar o ângulo de giro do seu eixo. Por sua facilidade de controle, esses dispositivos são bastante utilizados em projetos robóticos.

Apesar dos desafios encontrados durante a execução, o trabalho obteve êxito. Consideramos que o trabalho foi um sucesso, pois conseguimos implementar todas as demandas iniciais idealizadas na fase de projeto. É importante destacar que superamos diversos obstáculos na implementação do sistema, o que nos possibilitou um incremento positivo de conhecimento para nosso futuro acadêmico.

Este trabalho apresentou uma contribuição significativa para o campo da robótica, proporcionando um sistema eficiente, seguro e acessível. Essa solução pode ser aplicada em diversas áreas, como indústria auxiliando na produção de peças para cada produto feito, educação (estimulando o aprendizado e o interesse nesse campo) ou até mesmo em viagens espaciais, onde os robôs partindo para lugares em que os seres humanos não podem alcançar nesse presente, sendo assim explorando áreas novas nesse imenso universo .

REFERÊNCIAS BIBLIOGRÁFICAS

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

CONTROLE de PS2 e Arduino. [*S. l.*], 10 set. 2015. Disponível em: https://tecnoinfomais.blogspot.com/2015/09/controle-de-ps2-e-arduino.html. Acesso em: 22 maio 2023.

DATASHEET DO 74HC148, 1992. Disponível em: https://pdf1.alldatasheet.com/datasheet-pdf/view/23041/STMICROELECTRONICS/74HC148.html. Acesso em: 22 maio 2023.

DICIONÁRIO OXFORD. Disponível em: https://languages.oup.com/google-dictionary-pt/. Último acesso: 21 de março de 2023.

DORF, Richard C.; SVOBODA, James A. **Introdução aos circuitos elétricos** . Grupo Gen-LTC, 2012.

GUIMARÃES, Fábio. Eletrônica: Como é um Joystick por dentro?. [*S. l.*], 3 ago. 2017. Disponível em: https://mundoprojetado.com.br/como-e-um-joystick-por-dentro/. Acesso em: 18 mar. 2023.

KIM, S. H. et al. Game Controller as a Low-Cost Input Device for Medical and Industrial Applications. IEEE Access, v. 8, p. 166792-166801, 2020. DOI: 10.1109/ACCESS.2020.3020199.

NUSSEY, John. **Arduino para leigos.** 2. ed. São Paulo: Alta Books, 2019.

PHENOPTIX. **MeArm V0.4 - Pocket Sized Robot Arm.** Disponível em: https://www.thingiverse.com/thing:360108. Acesso em: 23 maio 2023.

SOUZA, Joedson. e FERREIRA. Magnum. **DESENVOLVIMENTO DE UM BRAÇO ROBÓTICO COM CONTROLE MULTIMODAL USADO ESP8266 NODEMCU.** Orientador: Daniel dos Anjos Costa. 2023. 68 f. Trabalho de Conclusão de Curso (Técnico em Informática) - INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DA BAHIA, Euclides da Cunha-BA, 2023.