

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FEMEC - FACULDADE DE ENGENHARIA MECÂNICA

THIAGO AUGUSTO DE SANTANA

**IMPLEMENTAÇÃO DA INTEGRAÇÃO PNRD E iPNRD PARA MUNDO DE
BLOCOS**

Uberlândia

2023

THIAGO AUGUSTO DE SANTANA

**IMPLEMENTAÇÃO DA INTEGRAÇÃO PNRD E iPNRD PARA MUNDO DE
BLOCOS**

Projeto de Fim de Curso apresentado à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica.

Área de concentração: Automação e Robótica.

Área de habilitação: Engenharia Mecatrônica

Orientador: Prof. Dr. José Jean-Paul Zanlucchi
de Souza Tavares

Uberlândia

2023

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

S232 Santana, Thiago Augusto de, 1996-
2023 Implementação da integração PNRD e iPNRD para mundo de blocos [recurso eletrônico] / Thiago Augusto de Santana. - 2023.

Orientador: José Jean-Paul Zanlucchi de Souza Tavares.
Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Uberlândia, Graduação em
Engenharia Mecatrônica.
Modo de acesso: Internet.
Inclui bibliografia.
Inclui ilustrações.

1. Mecatrônica. I. Tavares, José Jean-Paul Zanlucchi de Souza ,1972-, (Orient.). II. Universidade Federal de Uberlândia. Graduação em Engenharia Mecatrônica. III. Título.

CDU: 621.03

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

THIAGO AUGUSTO DE SANTANA

**IMPLEMENTAÇÃO DA INTEGRAÇÃO PNRD E iPNRD PARA MUNDO DE
BLOCOS**

Projeto de Fim de Curso apresentado à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica.

Área de concentração: Automação e Robótica.

Área de habilitação: Engenharia Mecatrônica

Orientador: Prof. Dr. José Jean-Paul Zanlucchi de Souza Tavares

Uberlândia, 6 de novembro de 2023

Banca Examinadora:

José Jean-Paul Zanlucchi de Souza Tavares – Prof. Dr. (UFU)

Roberto Mendes Finzi Neto – Prof. Dr. (UFU)

Leonardo Rosa Ribeiro da Silva – Prof. Dr. (UFU)

AGRADECIMENTOS

Aos meus pais e familiares, em especial à minha mãe Amelia Pereira Tiago, pelo amor e carinho, pelo incentivo e apoio incondicional em todos os momentos.

À Universidade Federal de Uberlândia pela oportunidade de realizar esse curso.

Aos amigos que fiz durante a graduação, principalmente aos que fazem parte da 21ª Turma de Engenharia Mecatrônica.

Ao professor Dr. José Jean-Paul Zanlucchi de Souza Tavares, agradeço por sua orientação inspiradora e incentivo constante, que me motivaram a buscar a excelência em meu trabalho, tornando-o valioso não apenas para os outros, mas também para o meu próprio desenvolvimento.

RESUMO

A Indústria 4.0, conhecida como a Quarta Revolução Industrial, abrange um extenso conjunto de tecnologias de ponta, tais como inteligência artificial, robótica, internet das coisas e computação em nuvem, nesse contexto, o uso da tecnologia RFID (*Radio Frequency Identification*) está ganhando crescente aplicação nos setores industriais e logísticos, embora seu uso seja, em grande parte, direcionado à identificação de itens. Neste estudo, é implementado um modelo com base nas redes de Petri, conforme proposto por Tavares e Saraiva em 2010. Esta rede de Petri é denominada PNRD (*Petri Net Inside RFID Database* ou Rede de Petri Inserida em Base de Dados RFID) e será aplicada no contexto do mundo de blocos com o objetivo de estabelecer um controle adaptativo para um robô cartesiano e três blocos. O procedimento envolve a criação de um espaço de Petri e um modelo PNRD e iPNRD (PNRD invertida) para cada agente, bem como a integração de suas informações. Essa integração possibilita que o robô cartesiano organize de forma autônoma os blocos, tentando obter o estado final desejado, previamente informado e que é verificado ao longo do processo a partir das informações armazenadas nas etiquetas RFID de cada bloco, onde se estabelece que as etiquetas RFID funcionem como uma base de dados local contendo informações sobre a identificação do bloco, o processo esperado (representado pela matriz de incidência da rede de Petri) e o estado atual e desejado do bloco. Isso é alcançado por meio da comparação de informações lógicas e físicas, o que resulta na geração de uma sequência de movimentos e na execução das ações correspondentes e propicia um controle adaptativo a falhas devido a ações externas, falhas essas que podem ou não inviabilizar atingir o objetivo final de todos os blocos.

Palavras-chave: RFID; PNRD; iPNRD; Rede de Petri; Robô Cartesiano; Arduino; Mundo de Blocos.

ABSTRACT

Industry 4.0, known as the Fourth Industrial Revolution, encompasses an extensive set of cutting-edge technologies, such as artificial intelligence, robotics, internet of things and cloud computing. In this context, the use of RFID (Radio Frequency Identification) technology is gaining ground. growing application in the industrial and logistics sectors, although its use is largely aimed at identifying items. In this study, a model based on Petri nets is implemented, as proposed by Tavares and Saraiva in 2010. This Petri net is called PNRD (Petri Net Inside RFID Database) and will be applied in the context of the block world with the objective of establishing adaptive control for a cartesian robot and three blocks. The procedure involves creating a Petri space and a PNRD and iPNRD model (inverted PNRD) for each agent, as well as integrating their information. This integration allows the Cartesian robot to autonomously organize the blocks, trying to obtain the desired final state, previously informed and which is verified throughout the process based on the information stored in the RFID tags of each block, where it is established that the RFID tags function as a local database containing information about the block identification, the expected process (represented by the Petri net incidence matrix) and the current and desired state of the block. This is achieved through the comparison of logical and physical information, which results in the generation of a sequence of movements and the execution of corresponding actions and provides adaptive control against failures due to external actions, failures that may or may not make it unfeasible. Reach the final goal of all blocks.

Keywords: RFID; PNRD; iPNRD; Petri net; Cartesian Robot; Arduino; Block World.

LISTA DE ILUSTRAÇÕES

Figura 1	Componentes de um sistema RFID	15
Figura 2	Modelos de etiquetas RFID	16
Figura 3	Leitor RFID PN532 utilizado no projeto	17
Figura 4	Exemplo de modelagem gráfica de um semáforo na RP	18
Figura 5	Associação dos termos da Equação 1 com os elementos onde serão armazenados na abordagem PNRD	22
Figura 6	Exemplo da PNRD na verificação de mancais	23
Figura 7	Associação dos termos da Equação 1 com os elementos onde serão armazenados na abordagem iPNRD	25
Figura 8	Arduino Mega 2560	26
Figura 9	Ramps 1.4 RepRap Shield	27
Figura 10	Exemplo de Robô Cartesiano	28
Figura 11	Exemplo da PNRD aplicada ao mundo de blocos	29
Figura 12	Matriz representando o espaço de trabalho discretizado	30
Figura 13	Robô cartesiano utilizado no trabalho	32
Figura 14	Fonte de alimentação do sistema	33
Figura 15	Conjunto Ramps 1.4/Arduino utilizado no projeto	34
Figura 16	Conexões para o conjunto Ramps 1.4/Arduino	34
Figura 17	Conexões para o conjunto Ramps 1.4/Arduino	35
Figura 18	Sensores fim de curso e orientação dos eixos	36
Figura 19	Extremidade final do robô, vista posterior	37
Figura 20	Extremidade final do robô, vista frontal	37
Figura 21	Espaço de trabalho do robô	38
Figura 22	Etapas básicas na execução do software	39
Figura 23	Modelo PNRD do bloco A	40
Figura 24	Modelo iPNRD para o mundo de blocos utilizado	42
Figura 25	Modelo do Espaço de Petri	42
Figura 26	Mapeamento entre PNRD, iPNRD e espaço de Petri	43
Figura 27	Exemplo de exceção de bloco em posição inicial incorreta	48

LISTA DE ABREVIATURAS E SIGLAS

UFU	Universidade Federal de Uberlândia
PFC	Projeto de Fim de Curso
RFID	Identificação por Rádio-Frequência
PNRD	Base de dados inseridas em RFID
iPNRD	PNRD invertida
id	Identificação
RP	Rede de Petri
NFC	Comunicação por Campo de Proximidade
IDE	Ambiente de Desenvolvimento Integrado

LISTA DE SÍMBOLOS

kHz	Quilohertz
GHz	Giga-hertz
V	Volts
P	Número finito de estados
T	Conjunto de transições
A	Conjunto de arcos direcionados
W	Função peso
M_0	Vetor de marcações no estado 0
M_k	Vetor de marcações
M_{k+n}	Vetor de marcações no estado $k+1$
A^t	Matriz de incidência
u_k	Vetor de disparos
T_k	Transição k , onde $k = 0, 1, 2, \dots, n$

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Objetivos	13
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
1.2	Justificativa	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	TECNOLOGIA RFID	15
2.1.1	Etiquetas	16
2.1.2	Leitores	17
2.2	REDES DE PETRI	18
2.2.1	Representação Gráfica	18
2.2.2	Conceito de disparo	19
2.2.3	Representação matemática	20
2.3	PNRD	22
2.4	PNRD INVERTIDA	24
2.5	ARDUINO e Ramps 1.4	25
2.6	Robô cartesiano	27
2.7	ESTUDO DE CASO: MUNDO DE BLOCOS	29
3	METODOLOGIA E DESENVOLVIMENTO	32
3.1	Sistema Físico	33
3.2	Software	40
3.2.1	Funcionamento da PNRD e iPNRD no programa	41
3.2.2	Código Geral	44
3.2.3	Exceções tratadas pelo Software	48
3.3	Testes	50
4	RESULTADOS E CONCLUSÕES	52
4.1	Trabalhos Futuros e Melhorias	53
5	REFERÊNCIAS BIBLIOGRÁFICAS	54
6	APÊNDICE A – ELEMENTOS DO MODELO IPNRD	56
7	APÊNDICE B – ELEMENTOS DO MODELO PNRD	58

1 INTRODUÇÃO

A Indústria 4.0 é um conceito que representa a automação industrial e a integração de diferentes tecnologias como inteligência artificial, robótica, internet das coisas e computação em nuvem com o objetivo de promover a digitalização das atividades industriais melhorando os processos e aumentando a produtividade. Dentro do que foi descrito, a tecnologia RFID (*Radio-Frequency Identification*) se encaixa perfeitamente, sendo ela uma tecnologia de conexão sem fios que faz transmissão de dados via radiofrequência. O padrão é usado em etiquetas eletrônicas para permitir rastreamento ou identificação de objetos à distância, desde dentro do chão de fábrica, até no controle de estoque, gestão da qualidade, gerenciamento do *Supply Chain* e mesmo segurança de itens dentro de um ponto de venda, as aplicações são diversas. A RFID é usada para construir uma “internet das coisas” – uma rede que permitiria às empresas rastrear mercadorias através da cadeia de abastecimento global e executar muitas aplicações simultaneamente (Violino, 2005). De acordo com o analista de pesquisa de mercado IDTechEx (Das, 2005), as vendas acumuladas de etiquetas RFID durante os 60 anos até o início de 2006 atingiram 2,4 bilhões, com 600 milhões de etiquetas sendo vendidas somente em 2005. Em 2006, previa-se que seriam necessários 1,3 mil milhões de etiquetas e 500 milhões de etiquetas inteligentes RFID numa série de áreas, tais como retalho, logística, animais e agricultura, serviços de biblioteca e equipamento militar. A RFID tornou-se uma área nova e excitante de desenvolvimento tecnológico e está recebendo cada vez mais atenção

A rede de Petri (RP), por sua vez, é uma técnica de modelagem que permite a representação de sistemas, especialmente sistemas concorrentes, utilizando como alicerce uma forte base matemática (MACIEL, LINS, CUNHA, 1996). Utilizada como uma ferramenta de análise e controle de sistemas discretos, ela é capaz de modelar casos de concorrência, paralelismo, assincronismo, conflitos e deadlocks (ZHANG, WANG, WU, 2016). Tal versatilidade faz com que as redes de Petri sejam aplicáveis na descrição e visualização de sistemas com fluxo de trabalho complexo, que é o caso de vários processos de produção da atualidade.

A combinação de redes de Petri juntamente com sistemas de controle RFID foi proposta por Tavares e Saraiva (2010), essa organização foi denominada de “Redes de Petri inseridas em base de dados RFID” ou PNRD. Com a adoção desse sistema as etiquetas passam a conter informações relacionadas ao estado do objeto no qual elas estão inseridas como o posicionamento de tal objeto. Dessa forma, as etiquetas se tornam um banco de dados

distribuído cuja atualização é efetuada em tempo de execução. A verificação constante dessas etiquetas permite a identificação automática de eventos inesperados, podendo por exemplo mostrar uma trajetória incorreta de algum objeto. A PNRD como foi desenvolvida atua em agentes passivos contendo as etiquetas RFID enquanto o leitor tende a estar em agentes ativos, tendo isso em mente foi desenvolvido por Fonseca e Tavares (2017) a PNRD invertida ou iPNRD que consiste na alteração dos locais que os elementos da PNRD atuam, logo as etiquetas passam a estar presentes nos agentes ativos e os leitores estarão em agentes passivos.

1.1 OBJETIVOS

1.1.1 Objetivo geral

O objetivo geral deste projeto é a implementação de redes de Petri (PNRD/iPNRD) no contexto de mundo de blocos para criar um controle adaptativo de um robô cartesiano e três blocos a partir da demonstração de um processo de empilhamento de blocos simulando assim um possível armazenamento de estoque. O intuito é demonstrar a efetividade do uso dos métodos da PNRD/iPNRD e sua aplicabilidade no meio industrial.

1.1.2 Objetivos específicos

Os objetivos específicos para o projeto em questão são:

- Estudo e aprimoramento em relação ao uso do robô cartesiano;
- Estudo do funcionamento da PNRD/iPNRD;
- Desenvolvimento dos códigos para o funcionamento do robô cartesiano;
- Desenvolvimento dos códigos para a aplicação da PNRD e iPNRD;
- Introdução de uma garra para movimentar os blocos;
- Aprimoramento do sistema para que todos os elementos operem em um único conjunto Arduino + Ramps;
- Projeto e impressão em impressora 3D dos blocos utilizados;
- Testes e correções do sistema;
- Registro em vídeo do funcionamento do sistema.

1.2 JUSTIFICATIVA

A tecnologia RFID está em constante evolução nos ambientes de produção industrial e logística. Portanto, a criação de um modelo que possibilite um controle mais preciso dos produtos e de seus respectivos processos é justificada. Isso pode resultar em um aumento do valor agregado aos processos de produção e logística, com benefícios como a redução de perdas, monitoramento mais eficaz e, em certos casos, a capacidade de corrigir erros automaticamente, como é caso no uso da PNRD/iPNRD.

Este trabalho apresenta uma dentre as diversas aplicabilidades da PNRD/iPNRD, demonstrando de forma simples a importância do uso de redes de Petri assim com elementos de tecnologia RFID quando empregados juntamente a elementos de robótica em ambientes indústrias. Para a realização deste trabalho diversas áreas de estudo do curso de engenharia mecatrônica foram utilizadas e aplicadas, onde foram aplicados conhecimentos das mais diversas disciplinas vistas durante o curso, como por exemplo instrumentação, eletrônica, sistemas embarcados, simulação de sistemas automatizados, algoritmos e programação de computadores, robótica e redes industriais.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será detalhado os elementos básicos que compõem esse projeto, se aprofundando de forma teórica no funcionamento de cada um deles. A princípio será explicado o sistema de identificação utilizados, em seguida é introduzido o conceito de redes de Petri que posteriormente se integra com o sistema de identificação gerando-se assim a PNRD e iPNRD que são a base fundamental desse trabalho e objeto principal de estudo, após isso tem-se um detalhamento na parte de controle e movimentação do projeto onde se é mostrado a confluência entre o uso do Arduino e de uma Ramps para o funcionamento do robô cartesiano e por fim o estudo de caso para o mundo de blocos que é abordado nesse trabalho.

2.1 TECNOLOGIA RFID

Nos últimos anos, os métodos de identificação automática ganharam ampla popularidade em diversas indústrias, incluindo serviços, logística de compras e distribuição, manufatura e sistemas de gerenciamento de materiais. Esses métodos de identificação automática são utilizados para obter informações sobre pessoas, animais, mercadorias e produtos em trânsito e possuem uma alta aplicabilidade em setores industriais. Dentre esses métodos tem-se como destaque a tecnologia de identificação por radiofrequência (RFID - *Radio Frequency Identification*), que possui algumas vantagens em comparação a outras tecnologias do mesmo ramo, como por exemplo menor tempo de identificação, menor ocorrência de falhas e maior controle na segurança e nos fluxos de informações.

Uma tecnologia que sempre é comparada a RFID são as etiquetas de códigos de barras que desencadearam uma revolução nos sistemas de identificação há algum tempo atrás, mas que estão a ser consideradas inadequadas num número crescente de casos. Códigos de barra podem ser extremamente baratos, mas o seu obstáculo é a sua baixa capacidade de armazenamento e o fato de que eles não podem ser reprogramados.

Conforme é mostrado na Figura 1, um sistema RFID sempre é composto por no mínimo dois componentes, um transponder que normalmente se encontra no objeto a ser monitorado e de um leitor/escritor que tende a estar em um agente ativo do sistema, mais a frente será visto que esta disposição dos elementos nem sempre se dará dessa forma. A tecnologia de identificação por radiofrequência, utiliza ondas eletromagnéticas entre 125 kHz e 5,8 GHz, para armazenar e ler informações de pequenos dispositivos chamados etiquetas ou tags, que nesse

caso são o transponder, o que permite que não haja nenhum tipo de contato entre as partes deste sistema.

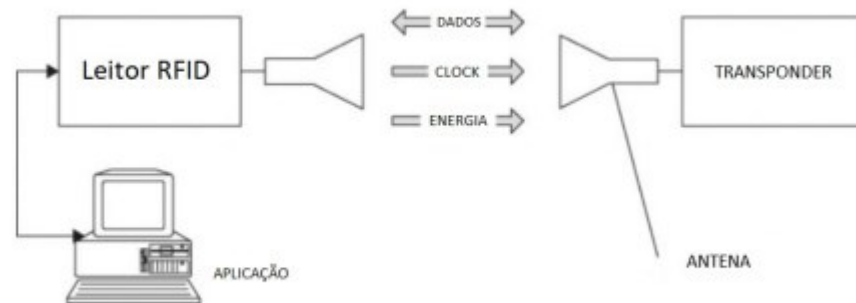


Figura 1 - Componentes de um sistema RFID

Fonte: FINKENZELLER (2010)

2.1.1 Etiquetas

A etiqueta (transponder) é o componente real de transporte de dados em um sistema RFID, tipicamente é composta por um elemento de acoplamento (antena) e um microchip eletrônico. A etiqueta normalmente não possui sua própria fonte de alimentação (como uma bateria), logo ela permanece passiva quando está fora do alcance de um leitor. Ela só se torna ativa quando está dentro da zona de alcance de um leitor, onde essa zona de alcance varia dependendo dos elementos utilizados. A energia necessária para ativá-la é fornecida por meio da unidade de acoplamento, sem utilizar nenhuma conexão por fio, juntamente com o pulso de temporização e os dados. Uma vez energizada é possível se realizar a leitura dos dados armazenados nela assim como a escrita de novas informações, podendo-se manter os dados anteriores ou alterá-los. Alguns exemplos de etiquetas que podem ser citados são as do tipo chaveiro, cartão e as do tipo folha adesiva, esses modelos de etiqueta podem ser vistos na Figura 2, onde os três modelos de etiquetas possuem frequência de 13,56 MHz, alcance efetivo de 7 centímetros e capacidade de armazenamento de 1Kbit. Para esse trabalho foi utilizado etiquetas com baixa frequência, diminuindo a distância de comunicação, onde pode-se estar mudando a potencia do leitor para se aumentar a frequência.



Figura 2 – Modelos de etiquetas RFID

Fonte: SILVA (2017)

2.1.2 Leitores

Os leitores, que, dependendo do design e da tecnologia utilizada, podem ser um dispositivo de leitura ou gravação/leitura, normalmente contém um módulo de radiofrequência (transmissor e receptor), uma unidade de controle e um elemento de acoplamento (antena) para com as etiquetas. Além disso, muitos leitores estão equipados com uma interface adicional para permitir o encaminhamento dos dados recebidos para outro sistema, que para este trabalho esses dados serão enviados a um Arduino. Um exemplo de leitor muito utilizado no mercado é do leitor RFID PN532 que pode ser visto na Figura 3, como visto ele possui diversas conexões além de possuir os elementos básicos de um leitor como já descrito.

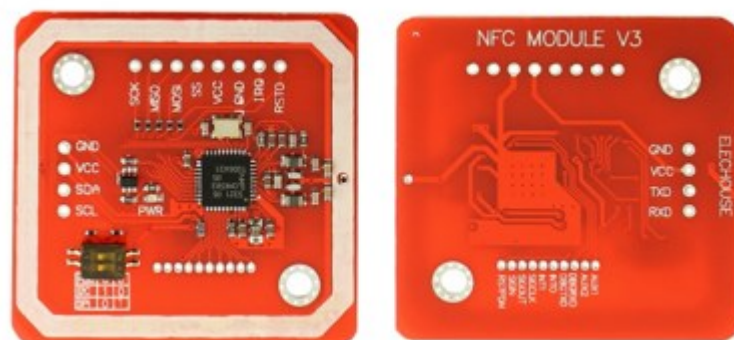


Figura 3 - Leitor RFID PN532 utilizado no projeto

Fonte: <https://www.usinainfo.com.br/1019342-thickbox_default/kit-leitor-rfid-nfc-tags-rfid-chaveiro-cartao-para-arduino-pn532.jpg> (Acesso em: 07/11/2023).

2.2 REDES DE PETRI

O conceito de redes de Petri (RP) tem a sua origem na dissertação de Carl Adam Petri, apresentada em 1962, e pode ser descrito como um modelo conceitual gráfico e matemático aplicável a muitos sistemas, que permite a verificação tanto das propriedades de um processo como o andamento do mesmo. Eles representam uma ferramenta altamente promissora para descrever e analisar sistemas de processamento de informações que apresentam características como simultaneidade, assincronia, distribuição, paralelismo, não determinismo e/ou estocasticidade (MURATA, 1989). Essas redes são aplicadas em uma ampla gama de domínios, incluindo controle de processos industriais, sistemas de comunicação, sistemas de software, protocolos de rede, logística e muito mais. A capacidade de representar sistemas complexos de forma visual e matemática torna as redes de Petri uma ferramenta poderosa para projetar e analisar sistemas e processos em diversas disciplinas.

2.2.1 Representação Gráfica

Como ferramenta gráfica, as redes de Petri podem ser usadas como um auxílio de comunicação visual semelhante a fluxogramas, diagramas de blocos e redes (MURATA, 1989).

Na modelagem gráfica, as RP são expressadas a partir de dois elementos visuais que são circunferências e traços ou retângulos, os quais representam lugares e transições respectivamente. Os lugares correspondem às variáveis de estado e as transições representam as ações ou eventos que acontecem no sistema. Em uma RP, lugares são ligados a transições por meio de arcos direcionados. Um arco obrigatoriamente liga um lugar a uma transição. Portanto, não é possível a ligação direta entre duas transições ou dois lugares, essa modelagem gráfica pode ser vista no exemplo da Figura 4, que demonstra o funcionamento de um semáforo de trânsito, no qual se tem representados três lugares e três transições. Há também um elemento presente na Figura 4 além dos lugares e transições, que é denominado marcação e é representado por um ponto presente dentro do lugar referenciado como “Verde”, essa marcação funciona como fichas ou tokens iniciais. Essas fichas indicam em que estado ou em que conjunto de estados o sistema modelado se encontra atualmente, mais a frente essas fichas também serão utilizadas no contexto de disparo. As marcações só podem estar presentes em lugares não existindo assim uma situação em uma transição possua uma marcação. Outro ponto importante também a se ressaltar é que o estado “Amarelo” não é indicado por uma transição, mas sim por um estado. A característica efêmera do estado amarelo pode dar a impressão que ele é uma

transição. No entanto, no modelo de redes de Petri, a transição é indivisível, ou seja, sem gradação visível.

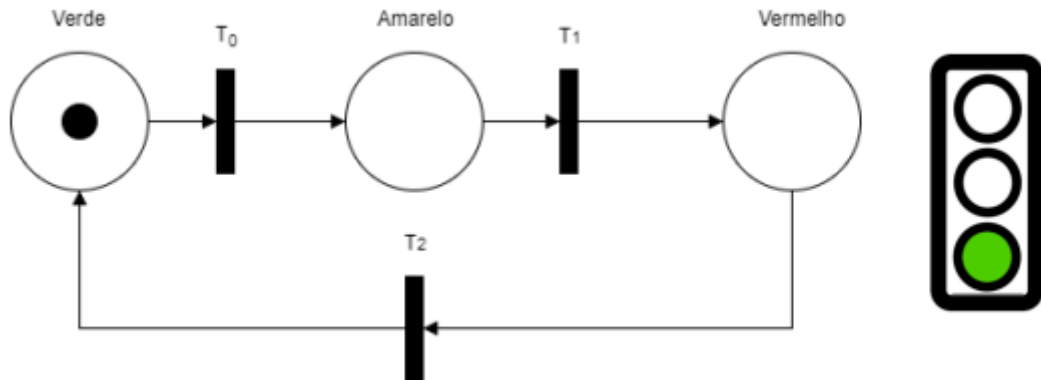


Figura 4 - Exemplo de modelagem gráfica de um semáforo na RP

Fonte: SILVA (2017)

Alguns conceitos precisam ser descritos nessa modelagem, que são os de entrada e saída de uma transição, onde se tem um arco que sai de uma transição e aponta para um lugar, o lugar para o qual o arco aponta será descrito como a saída da transição apontada pelo arco, esse caso pode-se ser notado na Figura 4, onde o lugar descrito como “Vermelho” é uma saída da transição T1, assim subsequentemente, se há um arco saindo de um lugar e entrando em uma transição, o lugar do qual o arco sai será uma entrada da transição a qual o arco aponta, como também pode-se ver na Figura 4, onde o lugar descrito como “Amarelo” é uma entrada da transição T1, sendo assim, a partir da Figura 4 pode-se ver que os lugares “Verde”, “Amarelo” e “Vermelho” são entradas para T0, T1 e T2 respectivamente, e também são saídas para T2, T0 e T1 respectivamente.

2.2.2 Conceito de disparo

Um disparo acontece quando uma transição ou um conjunto de transições é acionado de forma a alterar a marcação de uma RP. No exemplo do semáforo, pode-se disparar a transição T0 para alterar o estado do semáforo de verde para amarelo. Para que um disparo ocorra deve-se ter ao menos uma marcação no lugar de entrada da transição a ser acionada, caso não exista marcação ou marcações suficientes, o disparo é tido como falho ou que gerou uma exceção.

Em algumas redes de Petri os arcos entre transições e lugares possuem pesos. Se o peso estiver associado a uma entrada, ele indica a quantidade de fichas consumidas no disparo da

transição. Se ele estiver em uma saída, o mesmo indicará a quantidade de fichas criadas após o disparo. No restante desse trabalho, restringiremos as redes de Petri para redes com pesos unitários

2.2.3 Representação matemática

Como uma ferramenta matemática, nas redes de Petri é possível configurar equações de estado, equações algébricas e outros modelos matemáticos que governam o comportamento dos sistemas. Como descritor em (MURATA, 1989), uma rede de Petri pode ser definida como uma tupla de 5 elementos $PN = (P, T, A, W, M_0)$, onde:

$P = \{p_1, p_2, p_3, \dots, p_m\}$, é um conjunto finito não-nulo de estados.

$T = \{t_1, t_2, t_3, \dots, t_n\}$, é um conjunto finito não-nulo de transições.

$A \subseteq (P \times T) \cup (T \times P)$, é o conjunto de arcos direcionados de estado para transição e de transição para estado.

$W : A \rightarrow \{1, 2, 3, \dots\}$, é a função peso.

$M_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$ é a marcação inicial.

Uma vez definida a RP. Um disparo pode ser caracterizado pela Equação 1 a seguir:

$$M_{k+1} = M_k + A^T \times u_k \quad k = 0, 1, 2 \dots n \quad (1)$$

Em que:

M_k é chamado vetor de marcações e é um vetor coluna cujo elemento da n ésima linha corresponde a quantidade de marcações no n ésimo estado, em determinado instante de tempo k .

M_{k+1} é o vetor de marcações resultante do disparo k ;

A^T é a matriz de incidência, que representa o conjunto de arcos da RP. O elemento da matriz de incidência a_{ij} tem o valor de -1 quando o estado i for uma entrada da transição j e tem o valor de 1 quando for uma saída. Caso não exista relação entre o lugar e a transição, o elemento terá valor 0 . Nessa matriz o número de colunas é igual à quantidade de transições na RP e o número de linhas é igual à quantidade de lugares na RP. A matriz de incidência é de tamanho $m \times n$, sendo que n é o número de estados e m o número de transições na RP.

u_k é um vetor coluna cujo elemento da n ésima linha corresponde a quantidade vezes que a transição n foi realizada no disparo. Será aqui denominado vetor de disparos.

Utilizando a rede de Petri demonstrada na Figura 4, pode-se montar os elementos da Equação 1, onde se terá a matriz de incidência da seguinte forma:

$$A^t = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

Para se determinar o vetor de marcações M_k inicial, ou seja, M_0 , se tem que atribuir um ponto de início assim como visto na Figura 4, onde se tem uma marcação no lugar “Verde”, sendo assim se terá o seguinte vetor de marcações inicial para essa situação:

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Estabelecendo ainda que no instante 0 há o disparo da transição T0, que será responsável por dar início ao sistema, sendo assim, tem-se o seguinte vetor de disparos:

$$u_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Com todos os elementos determinados, a partir da Equação 1, tem-se que o resultado do disparo será o seguinte vetor de marcações M_1 :

$$M_1 = M_0 + A^t \times u_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

A partir da resolução da Equação 1 nota-se que o vetor marcação resultante é válido e que a marcação seguiu para o lugar seguinte, que é o lugar “Amarelo”, vale ressaltar que caso houvesse ocorrido o surgimento de números negativos a esse vetor de marcação, isso implicaria em uma exceção indicando que o processo não seguiu de forma correta e o vetor resultante seria inválido, logo a partir da Equação 1 se tem uma boa forma de checar exceções desse tipo.

2.3 PNRD

A PNRD é uma estrutura de dados formal a ser armazenada na etiqueta RFID de forma a integrá-la com sistema de controle (TAVARES e SARAIVA, 2010). Essa abordagem supõe que o processo experimentado pela etiqueta possa ser representado por meio de redes de Petri, de forma que o fluxo de trabalho da etiqueta seja descrito por uma matriz de incidência e seu estado atual seja representado por um vetor de marcações. Esses dois conjuntos de informações, juntamente com o ID da etiqueta e um indicador de tempo, compõem a estrutura típica da PNRD. A inclusão do indicador de tempo tem como objetivo permitir que o sistema realize análises quantitativas de desempenho. Os leitores, por sua vez, podem estar conectados a outros dispositivos de controle realizando os disparos, armazenando e fornecendo o vetor de disparos adequado a cada situação.

Como descrito a etiqueta RFID armazenara a matriz de incidência, o vetor de marcações e um indicador de tempo, assim como a etiqueta já consta com seu id pré-gravado, já o leitor que pode estar conectado a um dispositivo de controle, ficara responsável por conter e executar o vetor de disparos. Essa situação pode ser vista na representação da Figura 5.

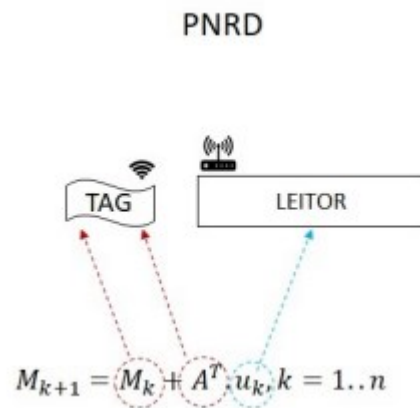


Figura 5 - Associação dos termos da Equação 1 com os elementos onde serão armazenados na abordagem PNRD

Fonte: FONSECA ET AL. (2018)

Uma característica importante da abordagem PNRD são as exceções. A PNRD é um ótimo recurso para o controle de processos, pois quando há a ocorrência de um erro, o mesmo é identificado imediatamente pelo leitor RFID, pois haverá o surgimento de um valor negativo no vetor de marcações.

Uma boa forma de exemplificar o uso da PNRD é por meio de uma situação cotidiana em indústrias, onde um produto deve seguir de um processo a outro, logo se fosse colocado uma etiqueta em cada produto a ser monitorado e cada etapa de processamento contasse com um leitor para tratar das informações das etiquetas, ao fim do processamento desse produto, um disparo poderia ser realizado, ativando-se assim uma transição que encaminharia o produto para a próxima etapa de processamento, caso o produto seguisse um caminho de forma incorreta seria gerado uma exceção, uma vez que o vetor de marcação, resultante da Equação 1, geraria valores negativos.

Para demonstrar de uma melhor forma a situação anteriormente descrita será utilizado um exemplo de uma fábrica que produza mancais, exemplo esse proposto por SILVA (2017), nesse exemplo que pode ser visto representado na Figura 6, tem-se uma situação em que após a usinagem de uma peça, é necessária a inspeção de dois tipos de requisitos: a cilindricidade e a rugosidade do seu furo. Caso o teste de cilindricidade falhar, a peça é enviada para a reciclagem. Se o teste de rugosidade falhar ela é colocada em uma fila para ser retrabalhada. Caso os dois testes estiverem de acordo com os padrões, a peça é armazenada em estoque.

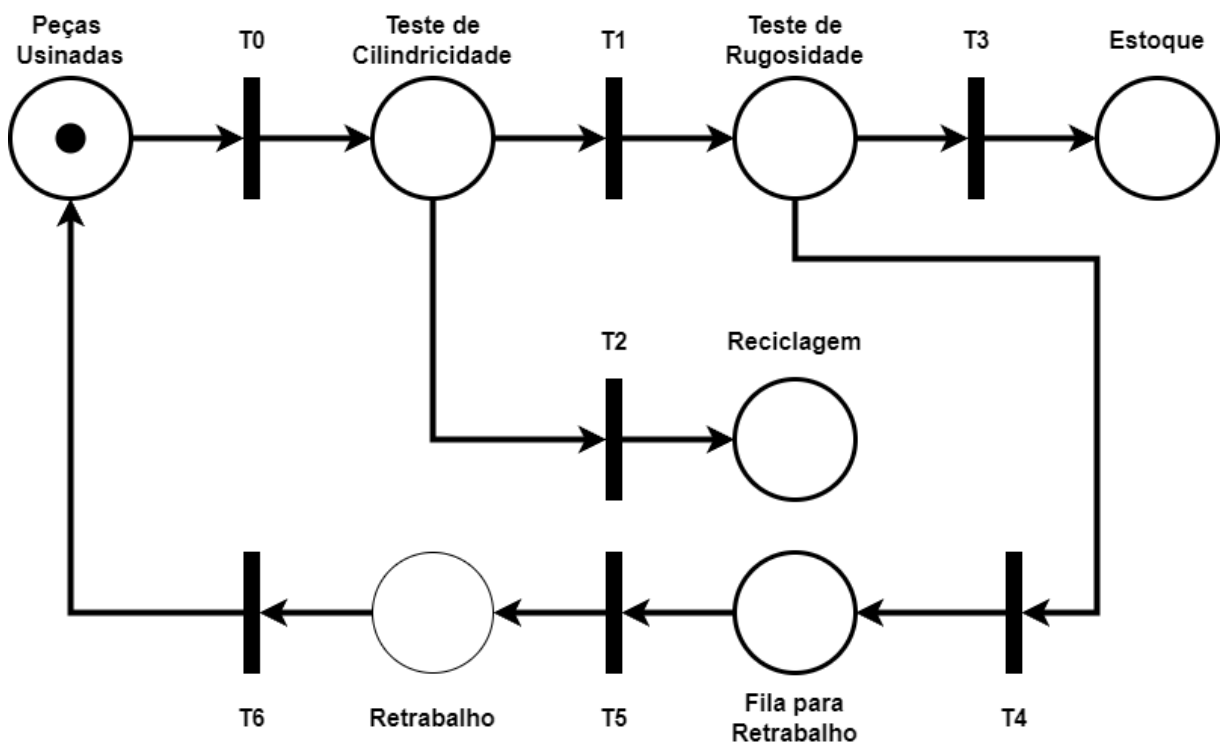


Figura 6 - Exemplo da PNRD na verificação de mancais

Fonte: SILVA (2017)

Para a utilização da PNRD, cada peça usinada teria que ser dotada de uma etiqueta RFID. Essa etiqueta deve conter a matriz de incidência da RP e o atual vetor de marcações, além de possuir um id de identificação. Em cada etapa as máquinas de verificação devem conter os leitores RFID necessários para a leitura das etiquetas.

Dessa forma, o leitor ligado a máquina de verificação de cilindricidade, ao perceber uma nova peça, adquire as informações contidas na etiqueta daquela peça, isto é, seu vetor marcação e sua matriz de incidência, realiza o disparo da transição T0, calcula o próximo estado e o grava na etiqueta. Em seguida, em concordância com o novo estado da etiqueta (teste de cilindricidade), envia um comando para começar o teste. Dependendo do resultado do teste, o sistema envia uma mensagem ao sistema de controle que possuem conexão com os leitores RFID, que por sua vez dispara a transição T1 ou a T2 e atualiza o novo estado da etiqueta, em seguida a peça é enviada para a reciclagem ou para o teste de rugosidade. Ambos os lugares são dotados de antenas RFID que leem a etiqueta e realizam um novo disparo. Se o resultado desse disparo for uma exceção, percebe-se que a peça não foi enviada para o lugar correto. De igual forma, pode-se detectar peças não-conformes ou não testadas mandadas para o estoque. Essa análise de exceções permite que a PNRD detecte e trate erros automaticamente em tempo de execução

Vale destacar que se é necessário um conjunto de informações, que podem ou não incluir: dados externos, marcação inicial, a identificação da etiqueta e a identificação da antena; para identificar uma transição disparada, sendo assim a mera passagem da etiqueta em um determinado leitor não define a transição disparada.

É importante destacar alguns pontos sobre a PNRD como sua escalabilidade dos processos em que ela atua, uma vez que o número de máquinas presentes no sistema não interfere no fluxo de trabalho, isso permite que o sistema seja capaz de crescer sem sofrer uma grande modificação no seu funcionamento. Também há agilização do fluxo de dados e diminuição dos custos de implementação de rede locais, uma vez que é a distribuição dos dados do processo é feita com o uso de etiquetas independentes, tornando desnecessária a consulta em um banco de dados central.

2.4 PNRD INVERTIDA

A criação da PNRD invertida (iPNRD) surge como uma variação da PNDR comum, onde a lógica de armazenamento das informações da Equação 1, é invertida. Como notado anteriormente a PNRD tem por objetivo a verificação de agentes passivos ao meio, como por

exemplo peças e produtos de um ambiente fabril, onde a matriz de incidência e vetor de marcação são armazenados nas etiquetas do agente passivo, enquanto as informações referentes ao vetor de disparos estão associadas aos leitores e sistemas de controle distribuídos no ambiente ou em um agente ativo. Em contra partida a isso a iPNRD funciona armazenando apenas o vetor de disparos nas etiquetas, enquanto a matriz de incidência juntamente com o vetor de marcação estão presentes no sistema que engloba o leitor e os controladores, como pode ser visto na Figura 7, dessa forma as etiquetas podem ser adicionadas a agentes ativos do meio, enquanto o leitor pode estar presente em agentes passivos.

Uma aplicação para a iPNRD foi proposta e desenvolvida por Fonseca e Tavares (2017), onde ela seria utilizada no contexto de busca e salvamento de pessoas em trilhas de caminhada, buscando-se assim viabilizar a utilização do conceito da PNRD em aplicações cuja a instalação de antenas RFID nos pontos de transição seja demasiadamente difícil.

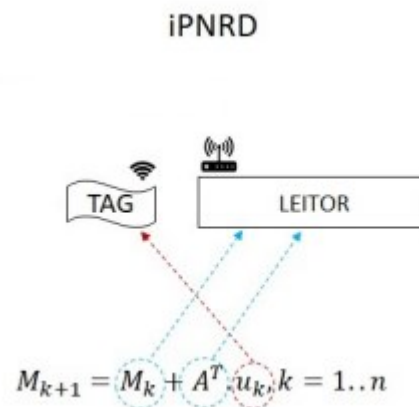


Figura 7 – Associação dos termos da Equação 1 com os elementos onde serão armazenados na abordagem iPNRD

Fonte: FONSECA ET AL. (2018)

2.5 ARDUINO E RAMPS 1.4

O Arduino é uma plataforma eletrônica aberta de fácil utilização. Ao longo dos anos, o Arduino tem sido o cérebro de milhares de projetos, desde objetos do cotidiano até instrumentos científicos complexos. Uma comunidade mundial de criadores - estudantes, amadores, artistas, programadores e profissionais - reuniu-se em torno desta plataforma de código aberto, e as suas contribuições somaram-se a uma quantidade incrível de conhecimento acessível que pode ser de grande ajuda tanto para novatos como para especialistas (ARDUINO, 2017). Ela possui as

características de um sistema embarcado multiuso, fornecendo suporte para a interação com diversos dispositivos eletrônicos, como LEDs, sensores, placas de comunicação, drivers de motores e leitores RFID.

Um dos motivos em se utilizar as placas Arduino se deve a alguns fatores como por exemplo ao fato delas serem relativamente baratas em comparação com outras plataformas de microcontroladores, seu ambiente de programação (IDE) é simples e claro além de ser executável em diversos sistemas operacionais, possui software e hardware de código aberto e extensível. A Figura 8 mostra uma placa Arduino mega 2560 contendo toda as suas conexões e componentes.

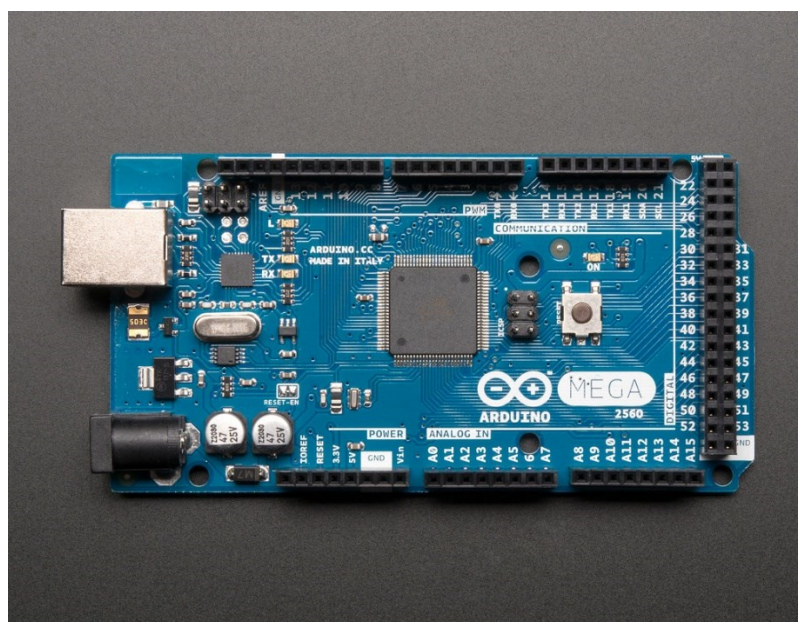


Figura 8 - Arduino Mega 2560

Fonte: <https://live.staticflickr.com/3741/9572391998_0d8f5614a0_b.jpg> (Acesso em: 08/11/2023).

Uma Ramps 1.4, ou *RepRap Arduino Mega Pololu Shield 1.4*, é uma placa eletrônica de controle projetada principalmente para uso em impressoras 3D de código aberto, como as impressoras RepRap. Ela atua como um controlador para diversos componentes de uma impressora 3D, incluindo os motores de passo, o aquecimento da cama de impressão e do extrusor, sensores e outros dispositivos relacionados ao funcionamento da impressora.

A Ramps 1.4 é compatível com placas Arduino Mega e é frequentemente usada com firmware personalizado ou código de controle para operar a impressora 3D. Ela fornece

conectores e interfaces para conectar e controlar os diferentes elementos da impressora 3D, permitindo movimentação precisa dos motores, regulação de temperatura, e aquisição de dados para monitoramento do processo de impressão.

Essa placa se tornou popular devido à sua versatilidade e ao código aberto, o que permitiu a comunidade de entusiastas de impressão 3D aprimorar e personalizar seus sistemas. Ela é uma parte fundamental de muitos projetos de impressão 3D devido à sua capacidade de fornecer controle preciso e flexibilidade na operação das impressoras e também pode ser utilizada para o controle de robôs do tipo cartesiano, uma vez que esses robôs operam da mesma forma que as impressoras 3D (REPRAP, 2023). Na Figura 9 esta demonstrada uma placa Ramps 1.4 com todas as suas conexões e componentes.

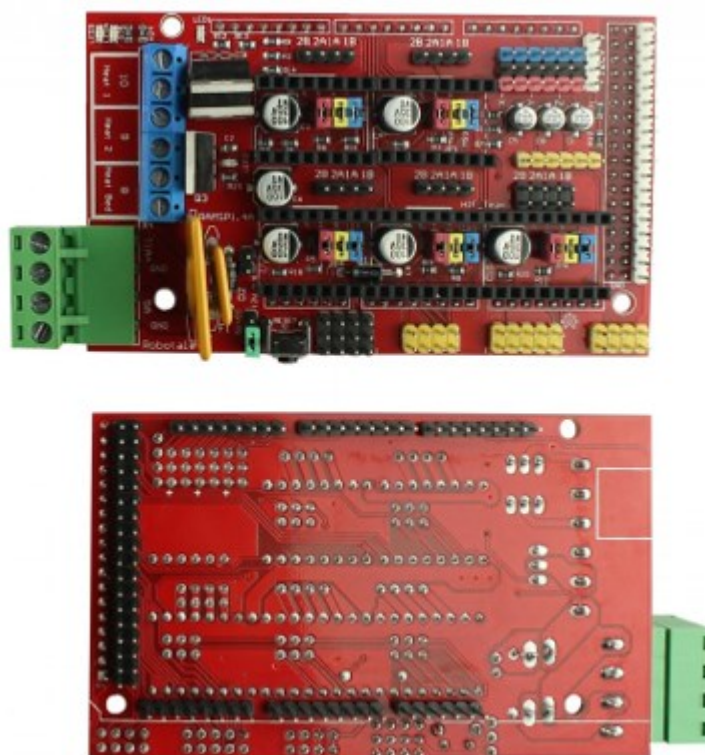


Figura 9 - Ramps 1.4 RepRap Shield

Fonte: <https://www.usinainfo.com.br/1023935-thickbox_default/ramps-14-reprap-shield-arduino-para-impressora-3d.jpg> (Acesso em: 08/11/2023).

2.6 ROBÔ CARTESIANO

Conforme pode ser visto na Figura 10, um robô de coordenadas cartesianas é um robô industrial cujos três eixos principais de controle são lineares (movem-se em linha reta ao invés

de rodarem) e estão em ângulos retos com relação a cada outro. Entre as suas vantagens, este arranjo mecânico simplifica o controle do braço. Robôs de coordenadas cartesianas com o membro horizontal suportado dos dois lados são geralmente chamados de Gantry robots.

Os robôs cartesianos possuem inúmeras vantagens, como a precisão de posição, operação simplificada, programação offline facilitada, alta personalização, boa capacidade de carga e menor custo em relação a outros robôs industriais tradicionais. As principais desvantagens deste modelo incluem a grande área operacional e de instalação necessárias para os robôs, o que encarece alguns projetos e inviabiliza outros (no caso de fábricas com linhas produtivas já existentes). Além disso, possuem uma montagem complexa (ROBÔS, 2023).

Esses robôs podem ser aplicados em tarefas como:

- Pegar e colocar (pick and place);
- Carregamento e descarregamento;
- Manuseio de materiais;
- Montagem e submontagem;
- Manipulação de materiais nucleares;
- Aplicações adesivas.

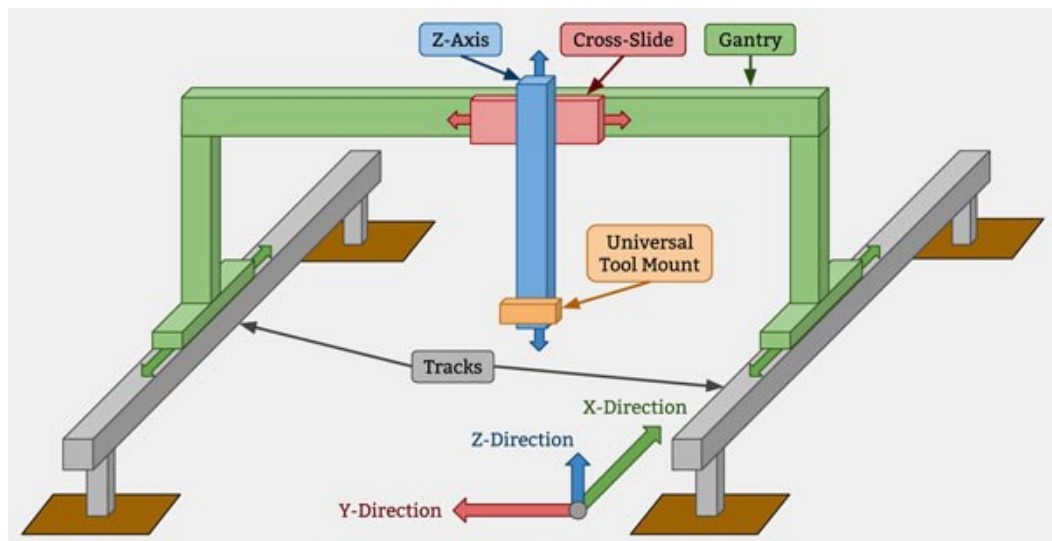


Figura 10 - Exemplo de Robô Cartesiano

Fonte: <https://www.venturus.org.br/wp-content/uploads/2019/07/1_QZexEQSiCs9Y5A4SZT5Bag-1.png> (Acesso em: 08/11/2023).

O robô cartesiano que será apresentado neste trabalho foi desenvolvido no trabalho de Oliveira (2020) e aperfeiçoado e utilizado no trabalho de Ugrinovich (2022).

2.7 ESTUDO DE CASO: MUNDO DE BLOCOS

Um dos mais famosos domínios de planejamento em inteligência artificial é o de mundo de blocos, que consiste em um conjunto de blocos sólidos colocados sobre uma mesa (ou, com maior frequência, sobre a simulação de uma mesa). Uma tarefa típica nesse mundo é reorganizar os blocos de certa maneira, utilizando um robô que pode erguer um bloco de cada vez. O objetivo é mover os blocos do estado inicial para alcançar o estado desejado. Apenas um bloco pode ser movido por vez: ele pode ser colocado na mesa ou colocado em cima de outro bloco. Por causa disso, quaisquer blocos que estejam em um determinado momento sob outro bloco não podem ser movidos. Um modelo físico é necessário e, devido aos limites, muitas configurações lógicas são impossíveis. Um exemplo da situação descrita pode ser visto na Figura 11.

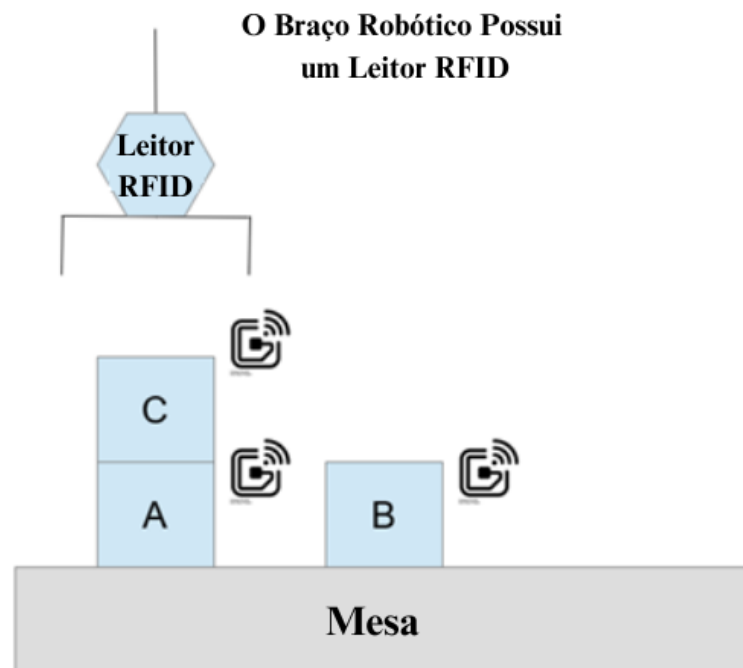


Figura 11 - Exemplo da PNRD aplicada ao mundo de blocos

Fonte: TAVARES e SOUZA (2019)

Neste trabalho em questão será trabalhado uma situação de mundo de blocos em que se terá três blocos que podem ser distribuídos por três colunas e três linhas, como está representado

na Figura 12, vale destacar que nessa imagem consta quatro linhas, isso se deve ao fato de que se é necessária uma linha a mais para a movimentação dos blocos para que os mesmos sejam movimentados sem o risco de ocorrer colisões com os outros blocos, mas apenas as três primeiras linhas serão ocupadas por blocos.

Para a situação apresentada, o número de possibilidades de posicionamento dos blocos seria de 60 posições distintas dentro desta matriz com os três blocos, tornando a complexidade do sistema demasiadamente extenuante, para se evitar um estudo desnecessariamente longo, uma vez que se quer apenas demonstrar o uso da PNRD e iPNRD, foi adotado regras limitantes para os blocos, como por exemplo que os bloco A, B e C só poderão se situar sobre a mesa nas colunas 1, 2 e 3 respectivamente, sendo assim o bloco A nunca estará sobre a mesa na coluna 2 e 3, e o mesmo vale para os demais blocos que não estarão sobre a mesa fora da coluna permitida, isso reduz o número de possibilidade para 13.

No decorrer do texto, é usada a seguinte notação para se referir ao posicionamento dos blocos:

- $AB_$ significa que o bloco A se encontra diretamente sobre o bloco B e B está sobre a mesa.
- $A_B_$ significa que os blocos A e B se encontram em colunas distintas, os dois estando sobre a mesa.
- $ABC_$ significa que A está sobre B, B está sobre C e C está sobre a mesa

O uso do underline () a frente da letra referente ao bloco indica que o mesmo está sobre a mesa. Portanto, se tem $BA_C_$, significa que o bloco B se encontra sobre o bloco A em uma coluna e o bloco A esta sobre a mesa, lembrando que o bloco A só pode estar sobre a mesa na coluna 1, sendo assim a partir dessa representação pode-se tirar de informação que tanto o bloco A quanto o bloco B se encontram na coluna 1, e o bloco C está isolado na coluna 3 também sobre a mesa. Todos os elementos presentes no modelo da PNRD e iPNRD estão discretizados nos apêndices B e C. Este exemplo está representado na Figura 12.

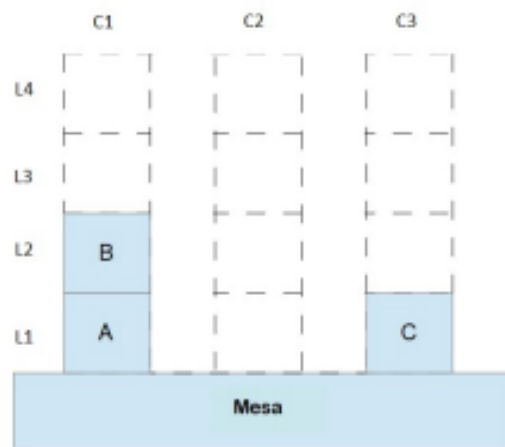


Figura 12 – Matriz representando o espaço de trabalho discretizado.

Fonte: TAVARES e SOUZA (2019)

3 METODOLOGIA E DESENVOLVIMENTO

Este trabalho tem por objetivo demonstrar o funcionamento e as vantagens do uso da integração da PNRD com a iPNRD aplicadas para um controle adaptativo no contexto do mundo dos blocos, destacando especialmente a identificação de exceções. Para ilustrar esse conceito, é apresentado um exemplo didático da PNRD juntamente com a iPNRD envolvendo três blocos e um robô cartesiano para a movimentação dos blocos.

O problema principal a ser tratado neste trabalho é o apresentado por TAVARES e SOUZA (2019), que foi descrito neste trabalho na seção 2, subseção 2.7 e pode ser visto na Figura 12, que consiste em três blocos distribuídos em um sistema 3x3, em que cada bloco possui uma etiqueta RFID anexada a sua superfície superior, sendo assim cada bloco terá um id próprio de identificação referente ao id exclusivo de cada etiqueta. O robô cartesiano será dotado de uma garra e um leitor/escritor RFID em sua extremidade final, a qual é responsável por identificar e gravar nas etiquetas e posteriormente agarrar e movimentar esses blocos para posições diferentes, essa extremidade com esses elementos que de agora em diante será mencionada como sendo o braço do robô.

Para a execução deste trabalho, foi desenvolvido tanto o sistema físico, contento os elementos do robô, do sistema RFID e dos módulos controladores, quanto a parte de software que é responsável pelo controle dos motores, controle da garra, controle do leitor RFID e operação das logicas de funcionamento aplicadas em conjunto com a parte de funcionamento da PNRD/iPNRD. Ao final, esse sistema deve ser capaz de identificar o arranjo inicial dos blocos e movimenta-los de tal forma que eles se encontrem nas posições finais desejadas formando um novo arranjo para esses blocos, sendo que ao decorrer de todo o processo todas as exceções estarão sendo checadas, informadas e tratadas. Vale também destacar que ao longo do processo toda a parte referente a PNRD/iPNRD estará sendo atualizada e checada, onde as etiquetas estarão sendo lidas e regravadas assim que necessário, mantendo atualizado toda as informações acerca dos blocos.

Ao longo deste capítulo são apresentadas todas as soluções implementadas neste projeto, e na Figura 13 está uma foto do sistema utilizado.



Figura 13 – Robô cartesiano utilizado no trabalho

Fonte: Registro local no laboratório (setembro de 2023).

3.1 SISTEMA FÍSICO

O sistema físico utilizado nesse trabalho possui elementos como:

- Placa mini controladora Arduino mega 2560;
- Notebook para utilização da IDE do Arduino;
- Placa Ramps 1.4;
- Fonte de alimentação ATX;
- Drivers A4988 para os motores de passo;
- Fios e jumpers para conexão;
- Estrutura do Robô cartesiano;
- Motores de passo Nema 17;
- Sensores fim de curso;
- Leitor RFID PN532;
- Garra Metálica;
- Blocos feitos em impressora 3D;
- Folha de orientação visual do posicionamento dos blocos;
- Servo motor.

O robô cartesiano utilizado neste trabalho foi desenvolvido no trabalho de Oliveira (2020), onde deve-se destacar que apesar de ter sido utilizado o robô desenvolvido em outro trabalho, ainda foi necessário realizar ajustes ao mesmo e também destacar que o desenvolvimento do robô não é o foco deste trabalho. O robô foi feito com tubos metálicos em sua estrutura e utilizando componentes impressos em impressora 3D, também conta com quatro motores de passo para o deslocamento nos eixos X, Y e Z. Para alimentar o sistema é utilizado uma fonte ATX, normalmente utilizada em computadores de mesa, ela é responsável por fornecer tensões de 12V para a Ramps 1.4 e de 5V para o servo motor que controla a garra, a fonte pode ser vista na Figura 14.

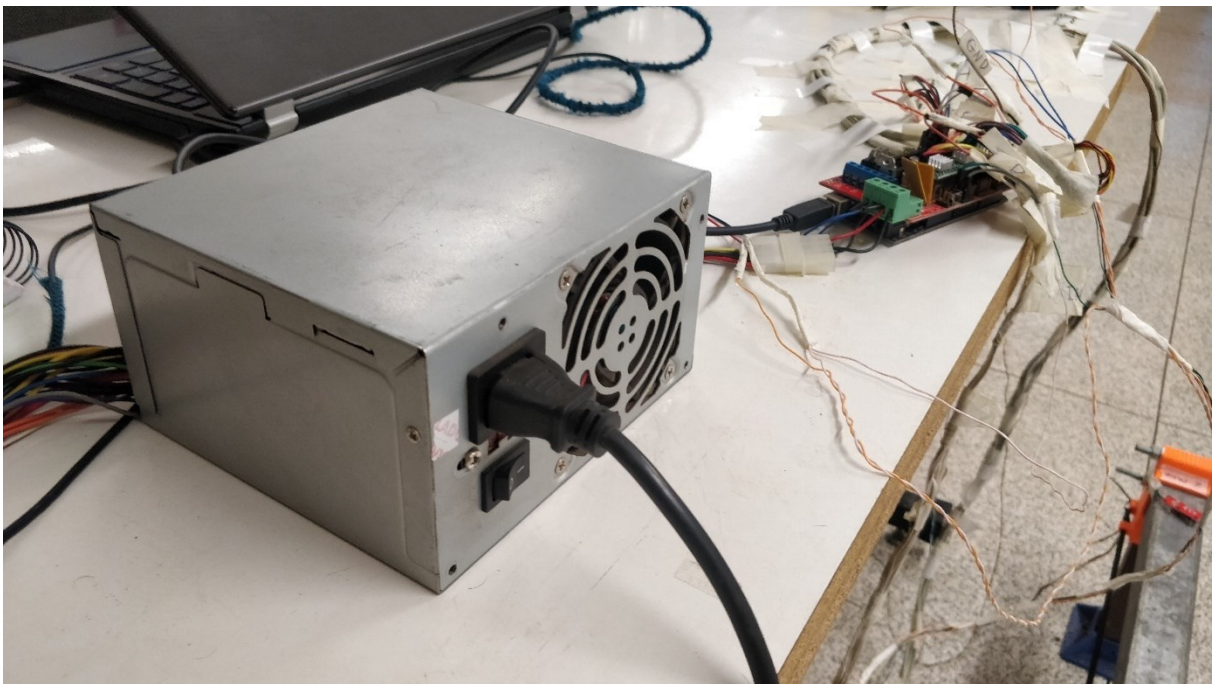


Figura 14 – Fonte de alimentação do sistema

Fonte: Registro local no laboratório (setembro de 2023).

A Ramps 1.4 por sua vez é conectada sobre o Arduino mega formando um conjunto para controle dos motores do robô, para tal também é necessário utilizar os drivers de motor que geram uma tensão correta para eles, já o Arduino é alimentado e recebe/fornece os dados através da conexão com o notebook. As conexões para o conjunto Arduino/Ramps podem ser vistas nas Figuras 15, 16 e 17, onde serão conectados os quatro motores de passo do robô já

com seus drivers, o controle do servo motor da garra, cinco sensores de fim de curso e o leitor RFID.

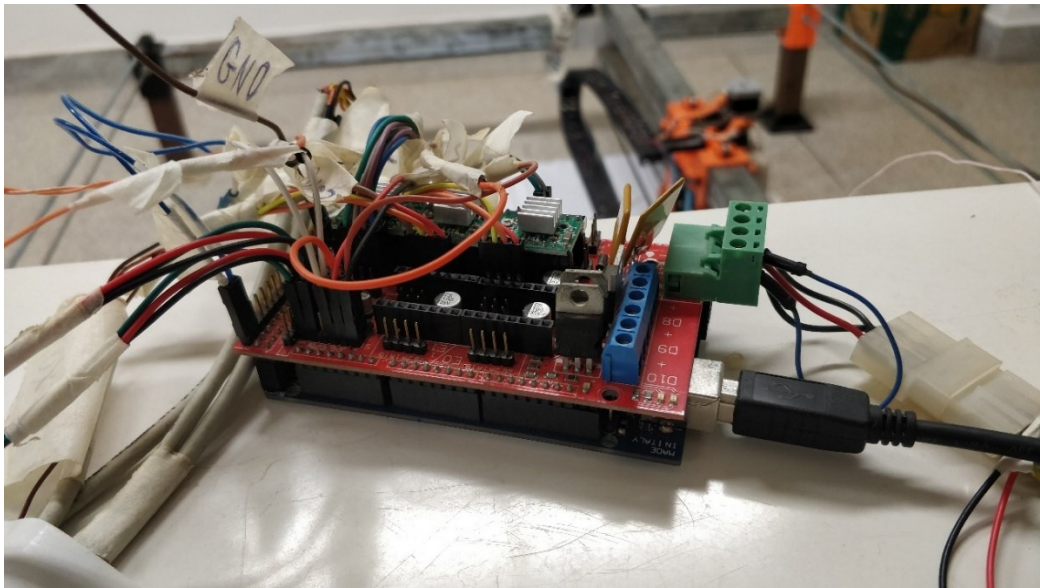


Figura 15 – Conjunto Ramps 1.4/Arduino utilizado no projeto

Fonte: Registro local no laboratório (setembro de 2023).

RAMPS 1.4 (RepRap Arduino MEGA Pololu Shield) GPL v3
reprap.org/wiki/RAMPS1.4

Reversing input power, and inserting stepper drivers incorrectly will destroy electronics.

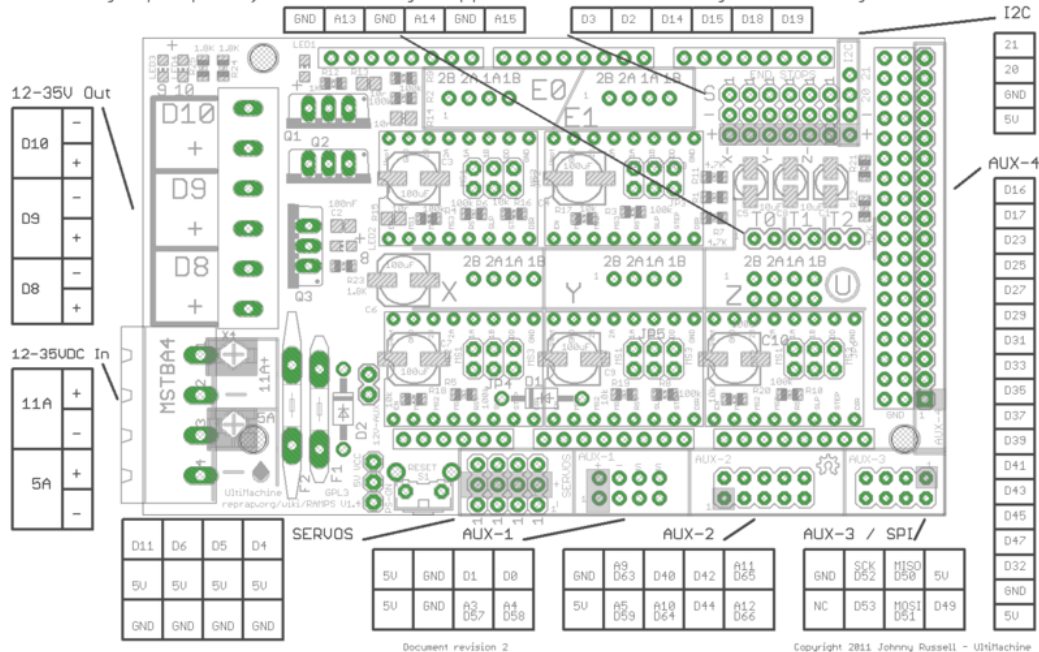


Figura 16 – Conexões para o conjunto Ramps 1.4/Arduino

Fonte: <<https://reprap.org/mediawiki/images/c/ca/Arduinomega1-4connectors.png>> (Acesso em: 11/11/2023).

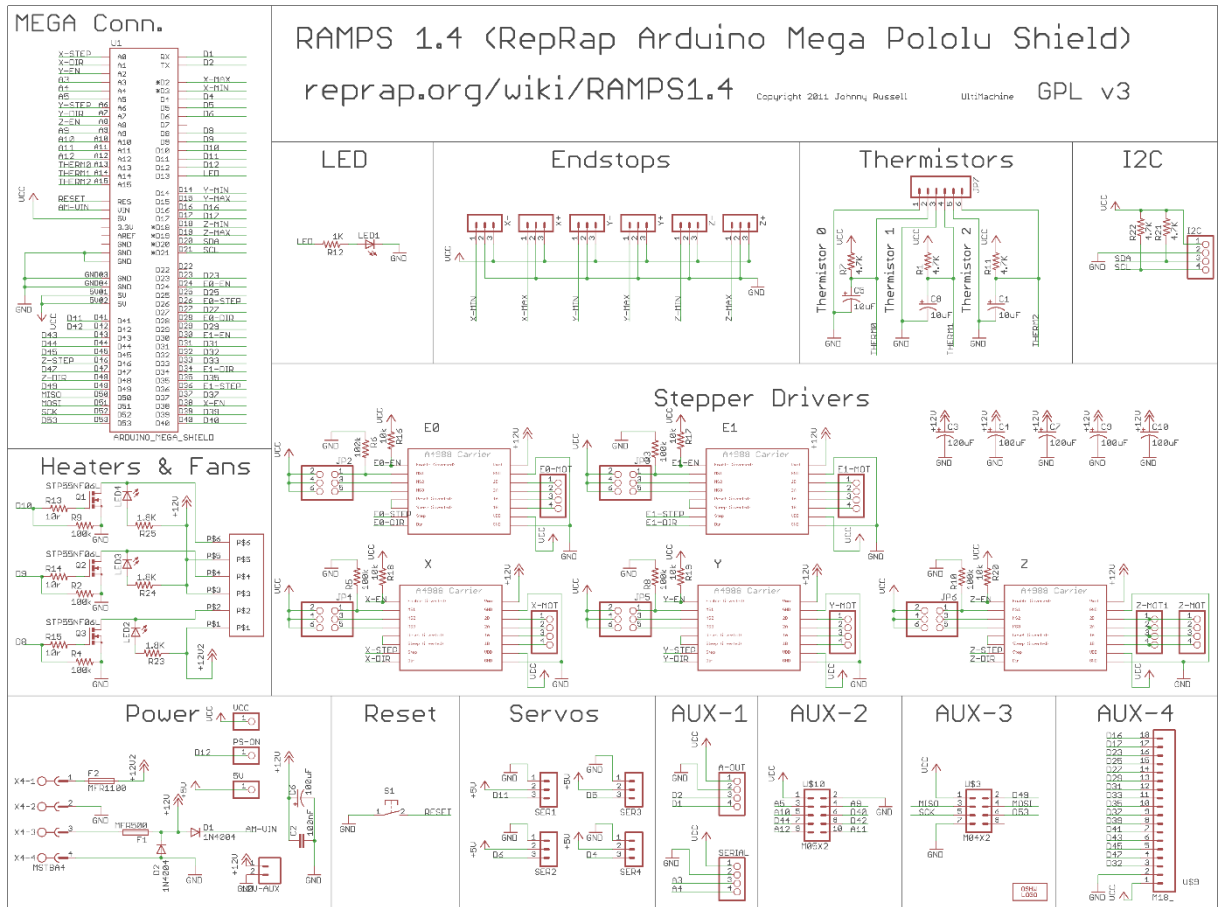


Figura 17 – Conexões para o conjunto Ramps 1.4/Arduino

Fonte: <<https://user-images.githubusercontent.com/19560798/41185260-bef28948-6b86-11e8-9929-f303c593b4ae.png>> (Acesso em: 11/11/2023).

Os sensores de fim de curso estão posicionados de tal forma a captar os sinais referentes as posições superiores e inferiores dos eixos X, Y e Z, a distribuição dos sensores assim como a orientação dos eixos esta apresentada na Figura 18, tal orientação foi adotada devido ao eixo Z da Ramps ser o único a controlar dois motores ao mesmo tempo, logo a direção que possui dois motores foi adotada como eixo Z, o eixo Y é referente a movimentação de subida e de descida da garra e leitor, e o eixo X é referente ao deslocamento restante. Como pode ser observado na Figura 18, os eixos possuem orientação negativa e positiva que servem para identificar qual sensor fim de curso foi acionado e em que ponto do eixo, esses sensores server tanto para se ter um referencial quanto para a segurança do sistema, evitando impactos ou que o motor funcione por tempo indeterminado. Um ponto importante a se destacar é que apenas o eixo Y não possui sensor fim de curso na parte inferior (-Y), pois não havia formas práticas de

se colocar ele, sendo assim toda sua limitação de movimentação para baixo se dá através do software, limitando a distância máxima que pode percorrer.

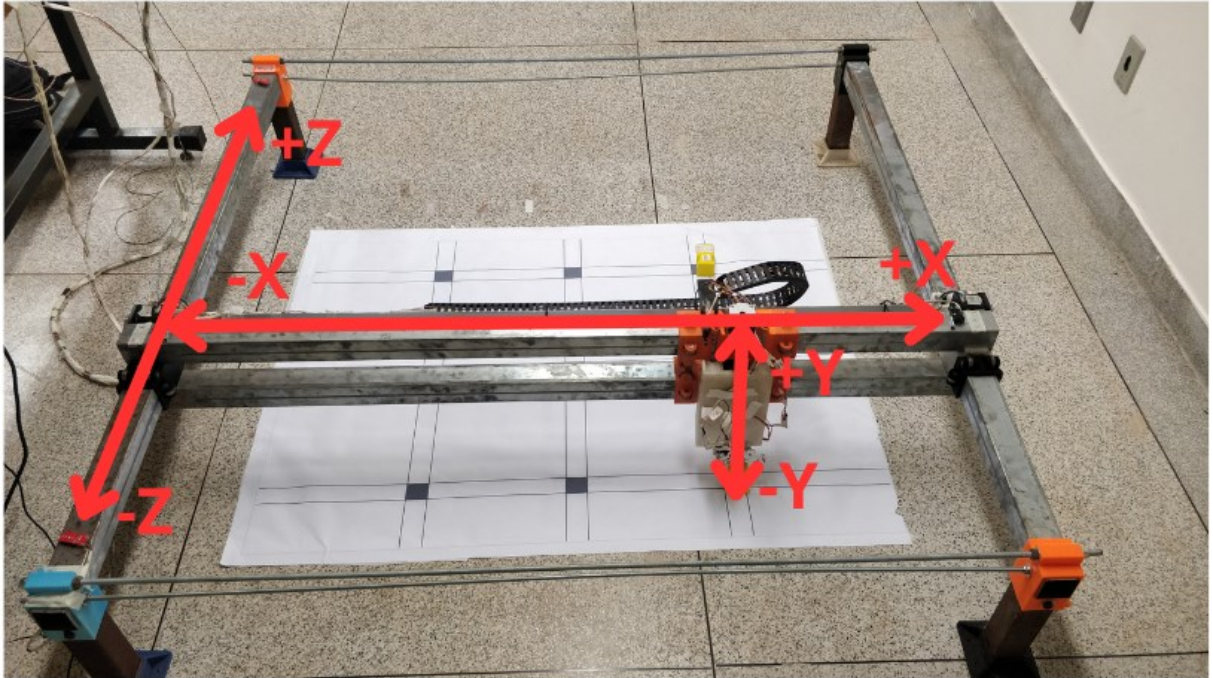


Figura 18 – Sensores fim de curso e orientação dos eixos

Fonte: Registro local no laboratório (setembro de 2023).

Na extremidade final do sistema, localizada na parte inferior do eixo Y (-Y), ocorre a instalação da garra e do sensor RFID, foi utilizado alguns suportes feitos em impressora 3D para o melhor posicionamento da garra juntamente com o servo responsável por abrir e fechar ela, assim como também posicionar o leitor RFID para ler a etiqueta presente no bloco, nas Figuras 19 e 20 pode-se ver essa distribuição. Além dos suportes para garra e leitor, os blocos utilizados também foram feitos em impressora 3D, eles foram projetados de tal forma a oferecer um bom encaixe e equilíbrio para a garra e um posicionamento adequado para a etiqueta.

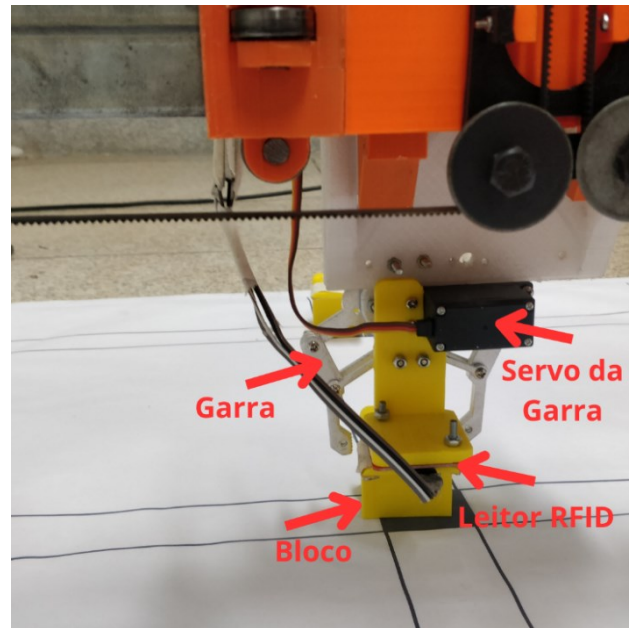


Figura 19 – Extremidade final do robô, vista posterior
Fonte: Registro local no laboratório (setembro de 2023).

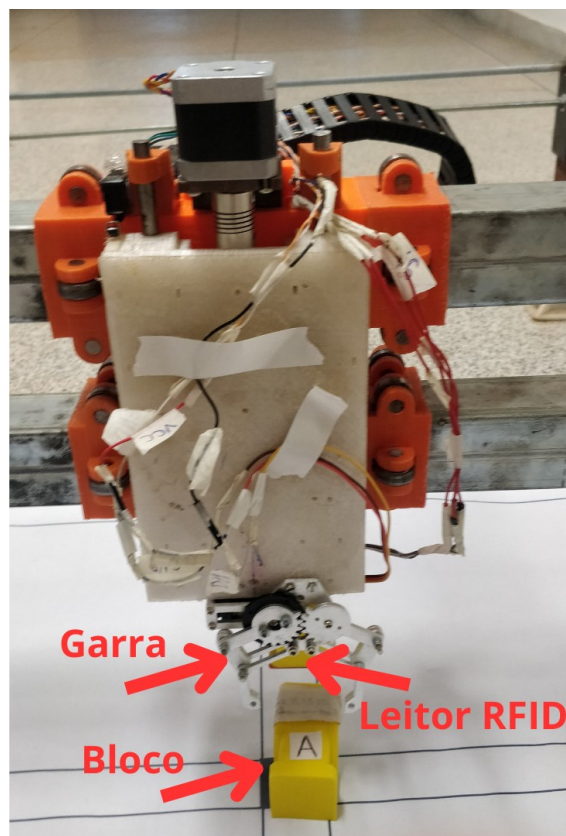


Figura 20 – Extremidade final do robô, vista frontal
Fonte: Registro local no laboratório (setembro de 2023).

O espaço de trabalho do robô foi determinado de tal forma que o mesmo percorresse a mesma distância em todas as direções, desta forma os locais de posicionamento dos blocos foram espaçados em 30 cm entre eles, formando um espaço de trabalho de 60x60 cm, essas dimensões são determinadas a partir da origem do sistema que se foi determinada como sendo a posição 1 (0x0), o espaço de trabalho pode ser visto na Figura 21. As posições foram expressas de duas formas, sendo elas numeradas de 1 a 9 ou em relação a Z e X, onde se terá Z-X com a origem em 0x0 ou posição 1, no trabalho foi utilizado uma folha de tamanho A0 para a orientação visual do posicionamento dos blocos.

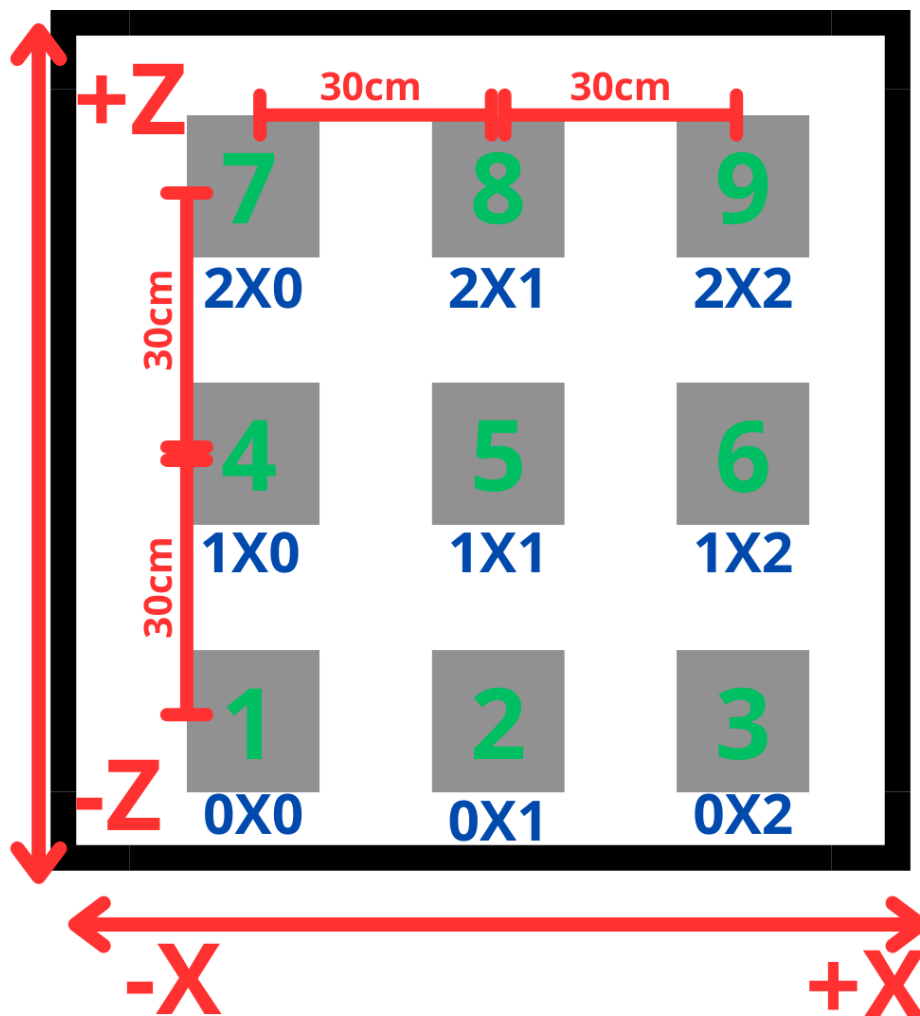


Figura 21 - Espaço de trabalho do robô

Fonte: Desenvolvido no site:< <https://www.canva.com/>> (Acesso em: 11/11/2023).

3.2 SOFTWARE

Para criar o software responsável por controlar todo o sistema e aplicar a PNRD e iPNRD, foi utilizado a IDE do Arduino, onde o código desenvolvido foi separado em funções individuais e posteriormente essas funções foram chamadas em um código principal, sendo elas executadas de forma encadeada. Nessa parte do trabalho será apresentado algumas das funções principais do código responsáveis pela lógica e funcionamento do sistema, aqui será apresentado o desenvolvimento e logica de funcionamento da PNRD e iPNRD que é o foco de estudo deste trabalho. Vale ressaltar que alguns nomes de variáveis, constantes ou funções dentro do código foram adotadas em sua maioria pelo autor deste trabalho, onde o mesmo buscou atribuir nomes que lhe fossem intuitivos, que não lhe causassem estranhamento, que fossem de fácil entendimento e lembráveis, então que se tenha em mente esses fatores caso algum nome pareça estranho, sempre lembrando que melhorias sempre poderão estar sendo feitas. A lógica básica de funcionamento do programa pode ser vista na Figura 22, onde estão destacadas as principais operações do código. As etapas presentes na Figura 22 serão abordadas e explicadas nas subseções 3.2.1, 3.2.2 e 3.2.3.

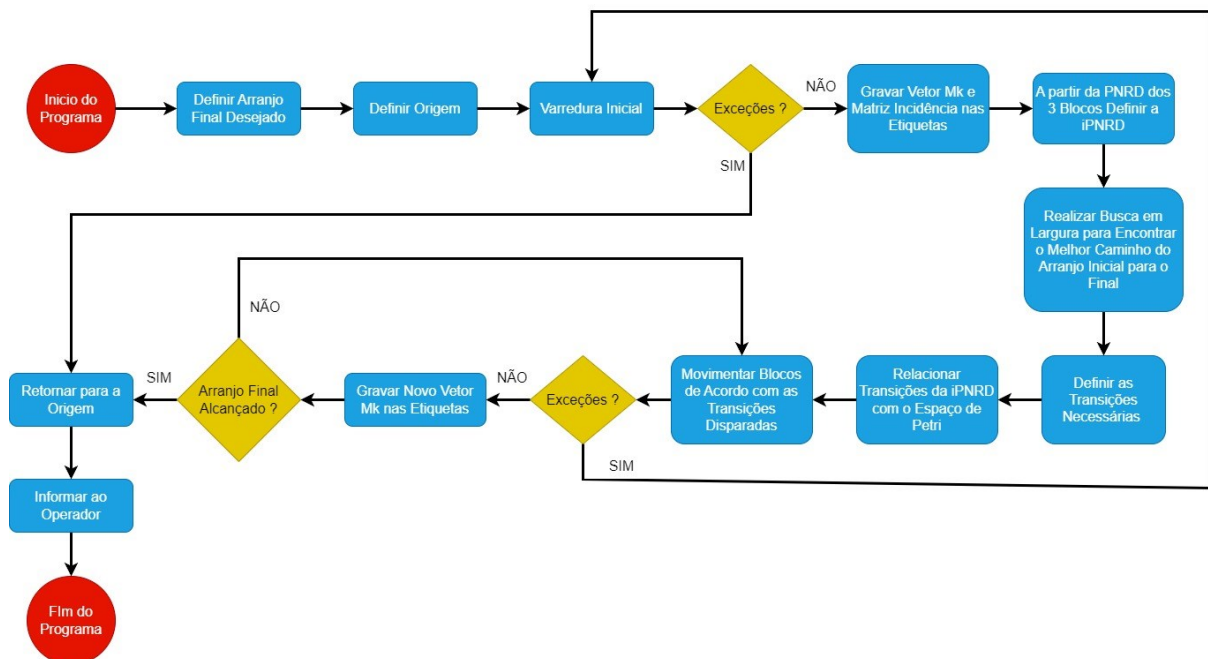


Figura 22 - Etapas básicas na execução do software

Fonte: Desenvolvido no site: <<https://app.diagrams.net/>> (Acesso em: 13/11/2023).

3.2.1 Funcionamento da PNRD e iPNRD no programa

Antes de se descrever o fluxo de funcionamento do código geral, será necessário explicar de forma mais detalhada como se dará o funcionamento da aplicação da PNRD e iPNRD no programa, onde para tal foi-se utilizado o trabalho de TAVARES e SOUZA (2019).

O sistema aqui tratado deve deslocar os blocos de forma a sair de um arranjo inicial e ir para um arranjo final, sendo assim, a primeira etapa a ser feita será determinar no código qual será o arranjo final desejado, já o arranjo inicial será obtido após uma varredura inicial que encontrara cada bloco no espaço de trabalho. Para se trabalhar com a PNRD/iPNRD se é necessário determinar os elementos da Equação 1, uma vez que já se sabe a modelagem do sistema, alguns desses elementos já podem ser descritos ao início do programa, como por exemplo a matriz de incidência, o vetor M_k será obtido após a varredura inicial e o vetor de disparo apenas após a determinação das transições necessárias para deslocar todos os blocos.

Durante a varredura inicial ocorrerá a verificação de cada bloco e a gravação de informações em suas etiquetas, nessa etapa do processo será descrita a PNRD que está sujeita ao posicionamento de cada bloco de forma individual, onde cada bloco terá sua própria rede de Petri que descrevera os possíveis arranjos que cada bloco pode ter, sendo assim, cada bloco terá sua rede de Petri própria que contará com sua própria matriz de incidência, seu vetor de marcações e seu vetor de disparo, um exemplo de rede de Petri usada na PNRD de cada bloco é a apresentada na Figura 23, onde pode-se ver a representação da PNRD do bloco A. Os elementos da Figura 23 estão descritos no Apêndice B.

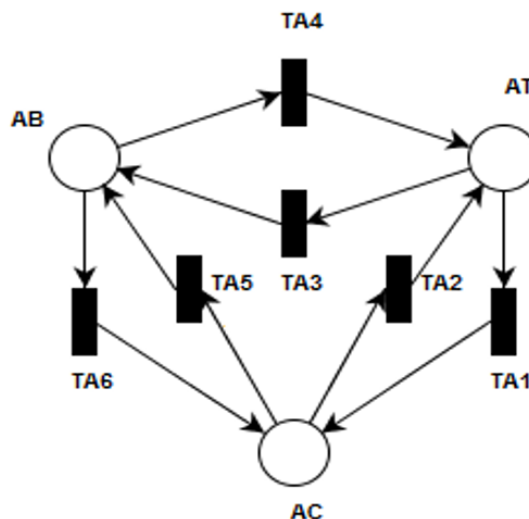


Figura 23 – Modelo PNRD do bloco A

Fonte: TAVARES e SOUZA (2019)

Um ponto importante a se dizer, é que apesar de cada bloco ter sua própria PNRD, elas acabam sendo idênticas, tendo a mesma distribuição, logo a PNRD do bloco B e C terão a mesma cara da PNRD do bloco A apresentada na Figura 23, dessa forma a matriz de incidência será idêntica para os três blocos e os vetores de marcação e disparo funcionaram de forma parecida, a matriz de incidência para a PNRD de cada bloco será a seguinte:

$$A^t = \begin{bmatrix} -1 & 1 & -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 & 1 & -1 \end{bmatrix}$$

Durante a varredura se é encontrado o posicionamento de cada bloco, com essa posição pode-se preencher o vetor de marcações do bloco naquele instante e assim gravar em sua etiqueta o vetor de marcações juntamente com a matriz de incidência, esse processo é feito para todos os blocos.

Com a PNRD de cada bloco definida, pode-se a partir das três, criar uma rede de Petri que descreva o modelo do problema do Mundo de Blocos, onde neste modelo estão contemplados todos os blocos, ou seja, esta é a perspectiva do agente ativo, sendo assim, estará sendo utilizado a iPNRD, diferentemente das redes de Petri de cada bloco, que se tratavam da perspectiva do agente passivo, fazendo-se assim o uso de uma PNRD. Neste modelo, cada estado se trata de um arranjo dentro do Mundo de Blocos, e as transições são mudanças de lugar de um único bloco, essa rede de Petri pode ser vista na Figura 24, os elementos contidos nela estão descritos no Apêndice A.

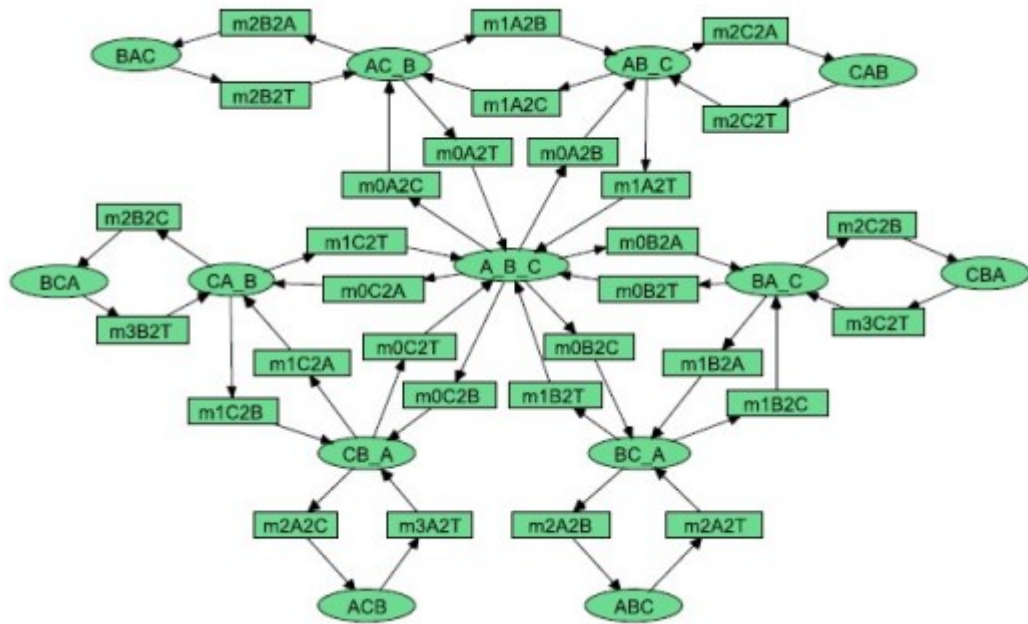


Figura 24 - Modelo iPNRD para o mundo de blocos utilizado

Fonte: TAVARES e SOUZA (2019)

Nesse ponto um outro elemento importante precisa ser descrito, se trata da movimentação do braço do robô no espaço físico. Para isto, foi modelada uma outra rede de Petri, representada na Figura 25, esta rede irá representar o espaço físico de trabalho. Neste sentido, as posições no espaço serão os estados desta rede de Petri. No decorrer do trabalho, esta rede será citada como espaço de Petri.

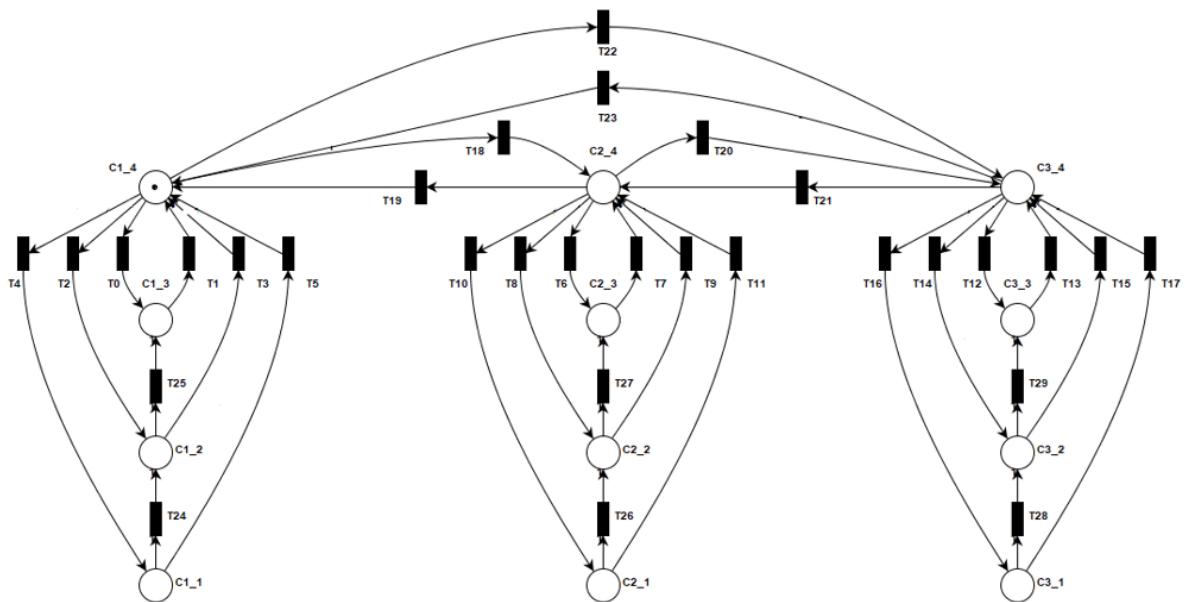


Figura 25 - Modelo do Espaço de Petri

Fonte: TAVARES e SOUZA (2019)

Na figura 26 seguir, se é apresentada a relação que pode ser feita entre a PNRD de cada bloco, a iPNRD do mundo de blocos e o espaço de Petri:

Espaço de Petri			PNRD			iPNRD
<i>A</i>	<i>B</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>C</i>	
<i>C1_1</i>	<i>C2_1</i>	<i>C3_1</i>	<i>AT</i>	<i>BT</i>	<i>CT</i>	<i>A_B_C</i>
<i>C1_1</i>	<i>C1_2</i>	<i>C3_1</i>	<i>AT</i>	<i>BA</i>	<i>CT</i>	<i>BA_C</i>
<i>C1_1</i>	<i>C2_1</i>	<i>C1_2</i>	<i>AT</i>	<i>BT</i>	<i>CA</i>	<i>CA_B</i>
<i>C1_1</i>	<i>C1_2</i>	<i>C1_3</i>	<i>AT</i>	<i>BA</i>	<i>CB</i>	<i>CBA</i>
<i>C1_1</i>	<i>C1_3</i>	<i>C1_2</i>	<i>AT</i>	<i>BC</i>	<i>CA</i>	<i>BCA</i>
<i>C2_2</i>	<i>C2_1</i>	<i>C3_1</i>	<i>AB</i>	<i>BT</i>	<i>CT</i>	<i>AB_C</i>
<i>C1_1</i>	<i>C2_1</i>	<i>C2_2</i>	<i>AT</i>	<i>BT</i>	<i>CB</i>	<i>CB_A</i>
<i>C2_2</i>	<i>C2_1</i>	<i>C2_3</i>	<i>AB</i>	<i>BT</i>	<i>CA</i>	<i>CAB</i>
<i>C2_3</i>	<i>C2_1</i>	<i>C2_2</i>	<i>AC</i>	<i>BT</i>	<i>CB</i>	<i>ACB</i>
<i>C3_2</i>	<i>C2_1</i>	<i>C3_1</i>	<i>AC</i>	<i>BT</i>	<i>CT</i>	<i>AC_B</i>
<i>C1_1</i>	<i>C3_2</i>	<i>C3_1</i>	<i>AT</i>	<i>BC</i>	<i>CT</i>	<i>BC_A</i>
<i>C3_2</i>	<i>C3_3</i>	<i>C3_1</i>	<i>AC</i>	<i>BA</i>	<i>CT</i>	<i>BAC</i>
<i>C3_3</i>	<i>C3_2</i>	<i>C3_1</i>	<i>AB</i>	<i>BC</i>	<i>CT</i>	<i>ABC</i>

Figura 26 - Mapeamento entre PNRD, iPNRD e espaço de Petri

Fonte: TAVARES e SOUZA (2019)

E é a partir dessas relações que se é desenvolvido a construção da PNRD, iPNRD e espaço de Petri no programa.

3.2.2 Código Geral

O código desenvolvido neste trabalho foi armazenado e disponibilizado de forma publica em um repositório do GitHub¹.

¹ Acesso ao GitHub contendo o código desenvolvido neste trabalho: <<https://github.com/MAPL-UFU/PNRD-iPNRD-Aplicada-ao-Mundo-de-Blocos-com-o-Robo-Cartesiano-.git>>

A princípio, com todo o sistema físico devidamente montado, a primeira parte do código se dá para a declaração e definição de algumas variáveis e constantes que serão usadas no decorrer do código, algumas foram declaradas de forma privada outras de forma global, dependendo de sua aplicação, mais a frente algumas configurações básicas são definidas. Para facilitar a elaboração do código foram utilizadas algumas bibliotecas, como a “AccelStepper.h”, que é responsável pelo controle dos motores, a “Servo.h” que controla servos motores e as bibliotecas “Pn532NfcReader.h”, “PN532_HSU.h” e “NfcAdapter.h” que servem para utilizar o leitor RFID.

Dentro do que se pode chamar de código principal, referente a parte de “void loop” do Arduino, se tem as chamadas das funções que em sua maioria estão condicionadas a uma série de fatores para que as mesmas possam executar. A primeira função a ser executada é a de determinar a origem do espaço de trabalho do robô, onde no código foi declarada com o nome de “zeragem”, esta função irá fazer com que todos os motores se movam, a partir de uma outra função responsável apenas pela movimentação dos motores, em uma direção até acionarem os sensores fim de curso de seus respectivos eixos, após acionados todos os sensores é definido um ponto de partida inicial, do qual irão partir até uma posição, que é definida no início do código, essa posição será a origem do espaço de trabalho e está referida como a posição 1 (0x0) como pode ser visto na Figura 21. A função “zeragem” é útil pois ao se iniciar o sistema, o robô não possui formas de saber em que posição ele se encontra, sempre atribuindo a origem ao seu primeiro local de inicialização, a partir dessa função, não importa onde o robô se encontre, ele sempre irá gerar uma origem assim como predeterminado, apesar que isso gera um gasto de tempo, pois toda vez que o sistema é ligado sempre será realizado a execução dessa tarefa., mas após isso se tem a origem e não se fara mais necessário o uso dessa função.

Após a definição da origem, o código entra em uma parte que será seu fluxo principal, onde foram aplicadas algumas condições de execução que lhe garantirão a possibilidade de ser reiniciado ou interrompido dependendo das exceções que podem ou não a vir ocorrer, lembrando que umas das partes de foco deste trabalho é exatamente a identificação e tratamento de exceções.

Tendo sido satisfeita a primeira condição, que garante que o código está passando pela primeira vez por aquela etapa e não que ele está sendo reinicializado, é necessário realizar uma varredura inicial para se encontrar todos os blocos e realizar a leitura e gravação nas etiquetas das informações referentes a eles, para tal é chamada a função de nome “varredura”, essa função irá percorrer todas as posições a procura dos blocos, a lógica de procura implementada aqui se

da seguinte forma: a varredura sempre partirá da esquerda para a direita começando pela origem (posição 1), caso haja bloco ali ele verifica a posição a cima desse bloco, pois se há bloco em baixo pode haver bloco a cima, caso não encontre ele irá para a posição sobre a mesa da coluna ao lado, pois se não há bloco ali não tem possibilidade física do bloco estar a cima, lembrando sempre que o modelo tratado simula um empilhamento de blocos só que representado no plano, dessa forma sempre haverá a necessidade de existir um bloco a baixo para que outro esteja a cima. Ao percorrer todas as posições necessárias e possíveis, o processo de varredura só irá se encerrar caso encontre todos os blocos ou caso ocorra uma exceção, tais exceções serão descritas e detalhadas em uma outra parte mais adiante.

O processo de varredura pode ser mais detalhadamente descrito da seguinte forma: ao finalizar a busca da origem, o braço do robô estará sobre a posição 1, logo ele será abaixado a partir de uma outra função responsável apenas por realizar os movimentos situados no eixo Y, uma vez abaixado será realizado uma tentativa de leitura, também a partir de uma outra função, caso seja identificado a presença de um dos blocos, o leitor/gravador irá gravar as informações iniciais daquele bloco, sendo elas o vetor M_k e a matriz de incidência A^t desse modelo, após a gravação, o braço do robô irá ser erguido e se encaminhará para a posição mais a cima daquele bloco, esse processo de movimentar, abaixar, ler, gravar e erguer será realizado seguindo a lógica descrita anteriormente. Tanto a matriz de incidência quanto o vetor M_k foram gravados em forma de string por ser a maneira mais fácil encontrada para se armazenar nas etiquetas, podendo ser convertidas para valores inteiros quando necessário.

Como descrito anteriormente, a partir da PNRD de cada bloco é possível determinar o modelo iPNRD para todo o sistema, sendo assim, com o uso de uma função chamada “pnrnToIPnrd”, é feita uma conversão a partir dos vetores M_k de cada bloco para se determinar como está o arranjo no modelo da iPNRD, assim como visto na Figura 26, o arranjo encontrado nesse momento se trata do ponto de partida para se alcançar o arranjo final desejado, como por exemplo pode-se ter inicialmente o arranjo BA_C_ e se desejar chegar ao arranjo BAC_.

Com o arranjo inicial e final definidos, o código segue para encontrar as transições necessárias para ir de um arranjo a outro, é desejável que o sistema encontre a solução que tenha o menor custo de tempo e de energia, logo, se estamos em um estado M_k e se quer ir para um estado M_{k+u} deve-se procurar o caminho que resulte na ativação do menor número de transições possível. As redes de Petri como dito anteriormente, são representadas por grafos direcionados, o que torna os mecanismos de busca em grafos uma escolha ideal para encontrar a solução ótima para o problema. Nesta parte do programa, foi desenvolvido um conjunto de funções com

o objetivo de se implementar o mecanismo de busca em largura, o qual consiste em se realizar uma busca partindo inicialmente dos estados adjacentes ao estado inicial sendo eles analisados primeiro, seguidos pelos estados adjacentes a eles. Essa análise segue uma abordagem de fila, em que os estados mais recentemente adicionados são os primeiros a serem examinados. A avaliação consiste, de maneira simples, em comparar esses estados com o estado desejado. Após encontrado o estado desejado, faz-se o caminho contrário, e então, são determinadas todas as transições que precisam ser ativadas para alcançá-lo.

Depois de se encontrar todas as transições iPNRD necessárias, o código terá um vetor de transições a serem acionadas na ordem em que estão posicionadas no vetor. Neste estágio, as transições da rede iPNRD estão identificadas, determinando quais blocos devem ser movidos e para onde. Contudo, é necessário ainda definir como essas transições serão executadas, isto é, como serão realizados os movimentos do braço do robô no espaço físico, para tal é utilizado o espaço de Petri, com isso é possível relacionar a ativação das transições com os movimentos físicos do braço do robô, onde com o uso da função “transicoesIPnrdToPetriSpace”, se é atribuído para cada transição da iPNRD uma sequência de transições do espaço de Petri, transições essas que estão diretamente associadas a uma determinada sequência de movimentos do robô, mas antes de irem diretamente para a movimentação, se é feita uma verificação a partir da função “condicaoInicialTransicoes” que tem por objetivo garantir que o braço do robô estará na posição necessária para dar início as transições do espaço de Petri. Com todos os requisitos atendidos as transições da iPNRD serão disparadas, disparando assim as transições do espaço de Petri que executara os movimentos necessários para deslocar os blocos.

Quando uma transição da iPNRD é disparada, intercalado com o processo de movimentação do robô, se tem um processo de leitura dos blocos e regravação de seu vetor M_k da PNRD, esse processo se dá da seguinte forma: ao se acionar uma transição, o braço do robô será movimentado e abaixado para cima do bloco a ser movido, nesse momento a etiqueta do bloco é lida para verificar se realmente se trata do bloco requerido, estando tudo certo, a garra é ativada e o bloco é preso a ela, em seguida o braço sobe e é movido juntamente com o bloco preso para a posição indicada pelas transições, o braço então é abaixado e se realizado uma nova verificação, que deve ler a etiqueta e verificar se não há a ocorrência de uma exceção, essa exceção em questão se trata do tipo que é identificado a partir da PNRD/iPNRD, o processo de identificação desta exceção será melhor descrito na subseção 3.2.3. Não havendo ocorrido nenhuma exceção, o vetor M_k do bloco é atualizado em sua etiqueta, a garra é acionada para

soltar o bloco, uma nova transição iPNRD é disparada e o braço do robô poderá seguir as mesmas etapas para o próximo bloco.

Ao fim de todo o processo, se não ocorreu nenhuma exceção os blocos serão deslocados na melhor ordem possível e ao final estarão nas posições desejadas, por fim o robô seguirá para a origem e encerrará o fluxo do programa.

3.2.3 Exceções tratadas pelo Software

Nesta parte do trabalho será explicado a ocorrência de algumas exceções que podem vir a ocorrer durante o funcionamento do sistema. As primeiras exceções a serem tratadas aqui estão relacionadas a condições fisicamente impossíveis ou não aceitas pela proposta deste trabalho e por fim será descrito as exceções identificadas pelo modelo PNRD/iPNRD.

A primeira exceção a ser identificada nesse programa, se dá na ocorrência do posicionamento fisicamente impossível de algum bloco², como por exemplo um bloco que está a cima de uma posição vazia, vendo isso na perspectiva de empilhamento de blocos, seria como se o bloco estivesse flutuando, o que logicamente seria impossível, mas como se está sendo simulado o empilhamento no plano, pode vir a ocorrer esse posicionamento incorreto. O posicionamento incorreto acarreta na mesma exceção da falta de bloco³, onde nos dois casos o bloco faltante ou o bloco indevidamente posicionado não serão identificados na varredura inicial, gerando uma exceção por falta de bloco, onde o tratamento dessa exceção será o retorno do robô para a origem e o encerramento do programa, que por final informará ao operador que há o problema de falta de bloco no sistema que deve ser resolvido antes de iniciar o programa novamente.

Outra exceção que pode acontecer é a de bloco repetido⁴, onde na varredura pode ocorrer de se identificar o mesmo bloco em mais de uma posição, isso pode vir a ocorrer caso algum agente externo modifique a posição de um bloco já identificado para um novo lugar ou caso as etiquetas tenham o mesmo id. Para essa exceção o tratamento é igual ao anterior, o robô retornará para a origem, o programa será finalizado e o operador será informado da inconsistência, sendo necessário resolvê-la primeiro para depois iniciar o programa.

² Vídeo demonstrativo disponível em: <<https://youtu.be/YfoOWGmTANg>> Acesso em 15/11/2023.

³ Vídeo demonstrativo disponível em: <<https://youtu.be/gi-WO20ZipM>> Acesso em 15/11/2023.

⁴ Vídeo demonstrativo disponível em: <<https://youtu.be/bbLRD7N9PVA>> Acesso em 15/11/2023.

Mais uma exceção⁵ que pode ocorrer é quando os blocos estão em posições nas quais eles não deveriam estar, como por exemplo o bloco A estar sobre a mesa na segunda coluna; como já foi definido anteriormente, o bloco A só poderá estar sobre a mesa na coluna 1, ou seja, na posição 1, o tratamento dessa exceção é exatamente o mesmo das anteriores, essa situação pode ser vista na Figura 27, onde o bloco A está na posição 2 e o bloco B está na posição 1.

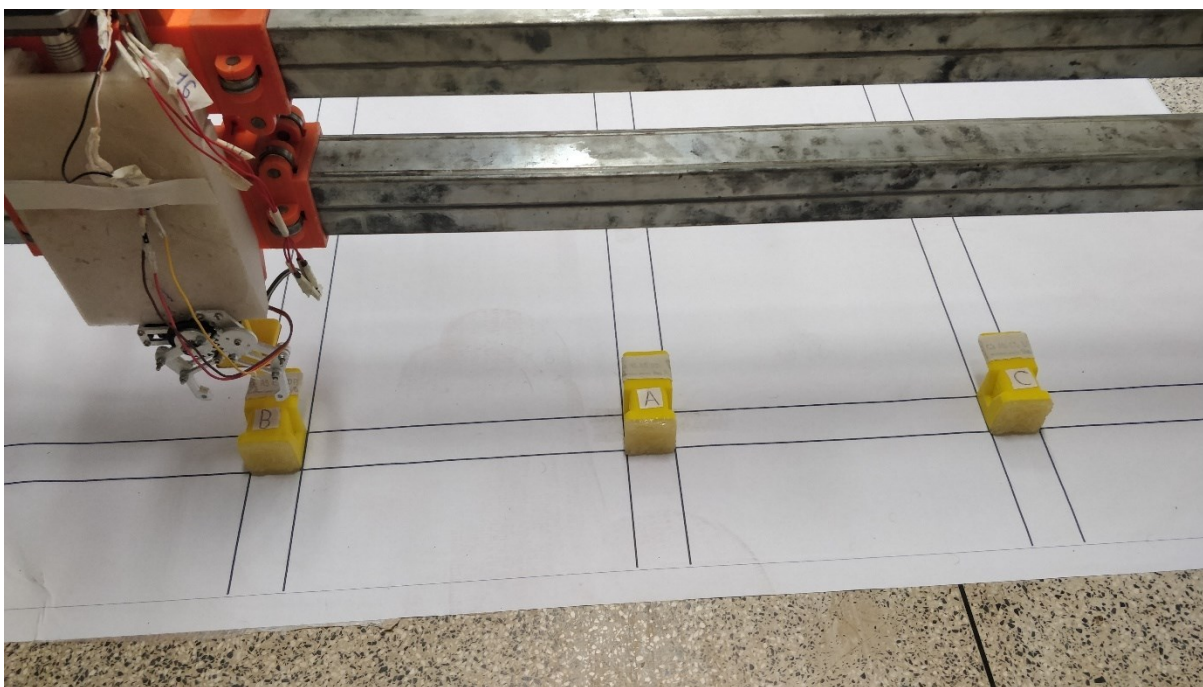


Figura 27 - Exemplo de exceção de bloco em posição inicial incorreta

Fonte: Registro local no laboratório (setembro de 2023).

Uma exceção bastante peculiar e que é tratada pelo programa, é a exceção que pode ocorrer após a varredura inicial⁶, onde já se teria encontrado o posicionamento de todos os blocos, mas ao se buscar um desses blocos para movê-lo o mesmo não estaria mais ali ou teria sido substituído por um agente externo. O tratamento dado a essa exceção é o de se reiniciar o programa a partir da varredura, pois o bloco pode apenas ter sido movido para uma posição diferente, sendo assim uma nova varredura já seria suficiente para determinar o novo posicionamento dos blocos e dar andamento ao programa. Um ponto importante a se destacar nessa abordagem, é que ao se reiniciar a partir da varredura inicial, todas exceções anteriores

⁵ Vídeo demonstrativo disponível em: <<https://youtu.be/DKvI2wck-yo>> Acesso em 15/11/2023.

⁶ Vídeo demonstrativo disponível em: <<https://youtu.be/OrAZ-LNu4U0>> Acesso em 15/11/2023.

serão novamente checadas e tratadas, evitando assim que a nova posição dos blocos implique em alguma outra exceção para o programa.

Por fim, a última exceção identificada e tratada nesse programa é justamente a que é foco desse estudo, se trata da exceção causada a partir da movimentação incorreta dos blocos que é identificada com o uso da PNRD/iPNRD⁷. O bloco em questão ao ser movimentado, seguiria por um caminho incorreto, onde no momento em que ele fosse ser deixado nessa posição o vetor M_k da PNRD gerado para essa posição estaria em desacordo com o vetor M_k encontrado pela Equação 1, indicando que o caminho que foi seguido não é o correto. O processo de identificação dessa exceção ocorre da seguinte forma: após se determinar as transições iPNRD que serão ativadas, pode-se construir o vetor de disparos da iPNRD e por consequência, os vetores de disparo para a PNRD de cada bloco, com isso já se tem todos os elementos da Equação 1, que retorna o próximo vetor de marcações para o bloco em análise naquele momento, sendo assim, tem-se a informação de para qual posição o bloco em questão deverá ir, isto é, o destino correto. Sempre que um bloco é movido, ao chegar em sua nova posição, o vetor de marcações em sua etiqueta deve ser atualizado com base na posição identifica pelo braço do robô, sendo assim, esse novo vetor poderá ser comparado com o obtido através da Equação 1, onde se os dois forem idênticos significara que o bloco seguiu para a posição correta, por outro lado, se forem diferentes significará que o bloco seguiu por um caminho incorreto. Esse tipo de exceção poderia ocorrer caso o robô apresentasse um mal funcionamento e seguisse de forma incorreta por um outro caminho, o mesmo ocorreria caso um agente externo forçasse essa alteração no caminho. O tratamento dado a essa exceção é o mesmo apresentado na exceção descrita anteriormente, implicando nas mesmas ocorrências já descritas.

3.3 TESTES

Para se comprovar o funcionamento de todo o sistema foram realizados vários testes, onde primeiramente se testou se o sistema estava realizando suas tarefas de forma correta sem a ocorrência de nenhuma exceção e posteriormente se verificou se estavam sendo detectadas as exceções e se o tratamento a elas estava ocorrendo conforme o planejado. Para se ter a ocorrência de exceções foi necessário forçá-las a ocorrer, para que assim pudessem ser realizados os testes.

⁷ Vídeo demonstrativo disponível em: <<https://youtu.be/Dhj59TAsARM>> Acesso em 15/11/2023.

Foram realizados três testes sem a ocorrência de exceções, consistindo em todos eles da distribuição dos blocos de um estado inicial para um estado final. Os arranjos finais e iniciais adotados nesses testes foram os seguintes:

- 1^o - Arranjo inicial BA_C_ deve ir para o arranjo final BAC_;⁸
- 2^o - Arranjo inicial CBA_ deve ir para o arranjo final BAC_;⁹
- 3^o - Arranjo inicial ACB_ deve ir para o arranjo final CBA_¹⁰.

Os testes foram executados e pode-se comprovar o perfeito funcionamento de todo o sistema, tanto físico quanto por parte do software. Para registrar e demonstrar a realização desses testes, foram gravados vídeos dessas operações, vídeos estes que estão disponíveis de forma pública no canal criado no site do YouTube para este trabalho (Projeto de PFC - Robô Cartesiano com PNRD E iPNRD, 2023).

Após os testes sem a ocorrência de exceções, foram realizados testes para se verificar a identificação e o tratamento de exceções, para isso foi preciso forçar a ocorrência delas, seja por se alterar os blocos fisicamente ou a partir de alterações no software. As exceções testadas foram as descritas na subseção 3.2.3 deste trabalho. Ao final de todos os testes pode-se constatar que todas as exceções foram identificadas e tratadas da forma desejada. Esses testes também foram gravados e disponibilizados no mesmo canal do YouTube trabalho (Projeto de PFC - Robô Cartesiano com PNRD E iPNRD, 2023).

O principal teste realizado que contém exceções, foi o teste de identificação de exceção por parte da PNRD/iPNRD, foco de estudo deste trabalho, onde o mesmo apresentou perfeito funcionamento. Para realizar este teste foram inseridos comandos no programa para que ao se movimentar o bloco B, o mesmo seguisse por um caminho incorreto, gerando assim uma discordância interna nos vetores M_k da PNRD do bloco, ao seguir por esse caminho incorreto, o programa acusou a exceção e seguiu para o devido tratamento dela, que nesse caso consiste em reiniciar o programa a partir da varredura inicial, por fim o sistema funcionou corretamente.

⁸ Vídeo demonstrativo disponível em: <<https://youtu.be/FA1FIgJyfMc>> Acesso em 15/11/2023.

⁹ Vídeo demonstrativo disponível em: <<https://youtu.be/XB5ZRhXz7rM>> Acesso em 15/11/2023.

¹⁰ Vídeo demonstrativo disponível em: <<https://youtu.be/WZlcO2WTGbo>> Acesso em 15/11/2023.

4 RESULTADOS E CONCLUSÕES

Ao final deste trabalho pode-se concluir que os estudos realizados alcançaram seus objetivos, demonstrando que todo o projeto obteve êxito em seu funcionamento, tanto o sistema físico quanto o software funcionaram corretamente. A utilização de um robô cartesiano para gerir um modelo de empilhamento de blocos utilizando-se das redes de Petri e de elementos RFID, resultou na demonstração da eficácia do uso da PNRD/iPNRD na gestão de exceções e controle de tráfego dos blocos.

Em síntese, a implementação bem-sucedida de elementos da Indústria 4.0 neste projeto, incluindo o aproveitamento de tecnologias RFID, do uso de um robô e microcontroladores, culminou na criação de um sistema altamente autônomo. Esse sistema, ao permitir que os próprios objetos (no caso, blocos etiquetados) determinassem suas ações, demonstrou efetivamente a capacidade de operar de maneira autônoma. A habilidade do sistema projetado em interpretar as informações, realizar o planejamento correspondente e operar de forma independente, sem intervenção humana, destaca o sucesso na aplicação prática das redes de Petri.

Vale destacar que durante o desenvolvimento, a parte de hardware apresentou um maior desafio para ser implementada, onde alguns elementos tiveram que ser adaptados e aperfeiçoados, elementos esses que acabaram ditando até mesmo o funcionamento do software. Devesse ter em mente essa questão para uma possível comercialização desse sistema, uma vez que o hardware demandará bem mais tratamento do que o software. O software por sua vez exigiu bem menos trabalho para funcionar corretamente, necessitando apenas de algumas melhorias para sua utilização comercial.

Por fim mas não menos importante, pode-se concluir que o trabalho realizado representa uma contribuição efetiva tanto para a formação do estudante envolvido quanto para o estímulo à realização de futuras pesquisas, uma vez que o desenvolvimento deste trabalho abrangeu diversas áreas de estudo do curso de engenharia mecatrônica, onde foram aplicados conhecimentos das mais diversas disciplinas vistas durante o curso, como por exemplo instrumentação, eletrônica, sistemas embarcados, simulação de sistemas automatizados, algoritmos e programação de computadores, robótica e redes industriais, além do aprofundamento em novos conceitos como o de tecnologias RFID e Redes de Petri.

4.1 TRABALHOS FUTUROS E MELHORIAS

Ao final deste trabalho, pode-se perceber as diversas possibilidades de aplicação que o uso de redes de Petri como a PNRD/iPNRD pode oferecer, tendo isso em mente alguns trabalhos futuros podem ser desenvolvidos a fim de se expandir as possibilidades de uso dessas redes, como por exemplo a aplicação para modelos mais complexos, no caso do mundo de blocos, modelos que possuam mais blocos ou outras condições de empilhamento, que tenham mais etapas, que necessitem de mais transições. Outros trabalhos poderiam ser desenvolvidos para simular as mais diversas situações encontradas em ambientes de fábrica, ou na logística de alguma empresa, com alta escalabilidade das redes de Petri as possibilidades de aplicação são diversas.

Apesar do ótimo funcionamento do sistema desenvolvido neste trabalho, foi possível notar algumas possíveis melhorias que poderiam ser aplicadas ao projeto, logo algumas sugestões de melhorias serão listadas a seguir:

- Utilizar fiação mais robusta para conectar os componentes;
- Testar mais tipos de etiquetas e leitores RFID;
- Armazenar e utilizar mais informações nas etiquetas;
- Desenvolver e testar um projeto para outros sistemas embarcados e outras linguagens de programação;
- Desenvolver uma garra mais eficiente para pegar os blocos;
- Projetar blocos mais eficientes;
- Aperfeiçoar o código desenvolvido neste trabalho;
- Aprimorar a mobilidade do robô de forma geral;
- Desenvolver um sistema de localização e de segurança para o robô, onde ao se iniciar o sistema, o robô já conseguisse identificar sua posição e ao se movimentar ele identificasse de forma mais precisa possíveis impactos no ambiente em que ele está.
- Desenvolver um sistema para detectar obstáculos no caminho do robô e para detectar se os blocos estão corretamente posicionados.
- Realizar estudos sobre interferência no sinal de comunicação do leitor e etiqueta para garantir a modulação de banda para que um sinal não interfira no outro.

5 REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINO. Arduino®. What is Arduino? Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em 08/11/2023.

ARDUINO. Site oficial do Arduino®. Disponível em: <<https://www.arduino.cc/>>. Acesso em 08/11/2023.

(FINKENZELLER, 2010) FINKENZELLER, K. RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication. [S.l.]: John Wiley & Sons, 2010. Citado nas páginas 14 e 15.

(FONSECA e TAVARES, 2017) FONSECA, J. P. S.; TAVARES, J.J.P.Z.S. Petri Net with RFID Distributed Database for aous Search and Rescue in Trails and Crossings. Proceedings of the International Workshop on Petri Nets and Software Engineering (PNSE'17), p. 229-230, 2017.

(FONSECA ET AL, 2018) FONSECA, J. P. d. S. et al. Redes de petri de alto nível e pnrnd invertida associadas ao controle de robôs móveis: uma abordagem para operações de busca e salvamento em trilhas e travessias. Universidade Federal de Uberlândia, 2018.

(MURATA, 1989) MURATA, T. 1989. Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, vol. 77, n° 4, p. 541-580.

(OLIVEIRA, 2020) OLIVEIRA, Nina Cervilha. Adaptação de robô cartesiano para máquina de corte a laser. 2020. 70 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Universidade Federal de Uberlândia, Uberlândia, 2020.

Pagina oficial da RepRap. Disponível em: <https://reprap.org/wiki/RAMPS_1.4>. Acesso em 08/11/2023.

(PIRES, 2020) PIRES, Alan Rogério Barreto. Sistema ciber-físico baseado na integração PNRD e iPNRD utilizando Cloud Computing. 2019. 73 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Universidade Federal de Uberlândia, Uberlândia, 2020.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Demonstração da exceção de Alteração no sistema pós Varredura. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/OrAZ-LNu4U0>>. Acesso em 15/11/2023.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Demonstração da exceção de ausência de algum Bloco. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/gi-WO20ZipM>>. Acesso em 15/11/2023.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Demonstração da exceção de Bloco em posição irregular. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/YfoQWGmTANG>>. Acesso em 15/11/2023.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Demonstração da exceção de Blocos Repetidos. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/bbLRD7N9PVA>>. Acesso em 15/11/2023.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Demonstração da exceção de posicionamento inicial incorreto dos Blocos. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/DKvI2wck-yo>>. Acesso em 15/11/2023.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Demonstração do funcionamento da PNRD e iPNRD. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/Dhj59TAsARM>>. Acesso em 15/11/2023.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Trajeto ACB_ para CBA_. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/WZlcQ2WTGbo>>. Acesso em 15/11/2023.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Trajeto BA_C_ para BAC_. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/FA1FIgJyfMc>>. Acesso em 15/11/2023.

PROJETO DE PFC - ROBÔ CARTESIANO COM PNRD E IPNRD. Trajeto CBA_ para BAC_. YouTube, 15 de novembro de 2023. Disponível em: <<https://youtu.be/XB5ZRhXz7rM>>. Acesso em 15/11/2023.

(ROBÔS, 2023) TIPOS DE ROBÔS INDUSTRIAIS E SUAS APLICAÇÕES. UNIVERSAL ROBOTS. Disponível em: <<https://www.universal-robots.com/br/blog/tipos-de-rob%C3%B4s-industriais-e-suas-aplicac%C3%B5es/>>. Acesso em 08/11/2023.

(SILVA, 2017) SILVA, Carlos Eduardo Alves da. Desenvolvimento de Biblioteca para Aplicações de PNRD e PNRD Invertida Embarcadas em Arduino. 2017. 101 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Universidade Federal de Uberlândia, Uberlândia, 2017.

(TAVARES e SARAIVA, 2010) TAVARES, J.J.P.Z.S., SARAIVA, T.A. Elementary Petri Nets inside RFID Database (PNRD). Journal International Journal of Production Research, vol. 48, 2010 – n. 9: RFID Technology and Applications in Production and Supply Chain Management.

(TAVARES e SOUZA, 2019) TAVARES, J. J.-P. Z. d. S.; SOUZA, G. d. A. Pnrd and ipnrd integration assisting adaptive control in block world domain. Proceedings of the International Workshop on Petri Nets and Software Engineering 2019, p. 73–90, 2019

(UGRINOVICH, 2022) UGRINOVICH, Vitor Santos. Adequação de robô cartesiano para trabalhar utilizando PNRD. 2022. 41 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Universidade Federal de Uberlândia, Uberlândia, 2022.

6 APÊNDICE A – ELEMENTOS DO MODELO IPNRD

Estados:

- BAC_ - Bloco B sobre A sobre C sobre a mesa.
- AC_B_ - Bloco A sobre C sobre a mesa e B sobre a mesa.
- AB_C_ - Bloco A sobre B sobre a mesa e C sobre a mesa.
- CAB_ - Bloco C sobre A sobre B sobre a mesa.
- BCA_ - Bloco B sobre C sobre A sobre a mesa.
- CA_B_ - Bloco C sobre A sobre a mesa e B sobre a mesa.
- A_B_C_ - A sobre a mesa, B sobre a mesa e C sobre a mesa.
- BA_C_ - Bloco B sobre A sobre a mesa e C sobre a mesa.
- CBA_ - Bloco C sobre B sobre A sobre a mesa.
- CB_A_ - Bloco C sobre B sobre a mesa e A sobre a mesa.
- BC_A_ - Bloco B sobre C sobre a mesa e A sobre a mesa.
- ACB_ - Bloco A sobre C sobre B sobre a mesa.
- ABC_ - Bloco A sobre B sobre C sobre a mesa.

Transições:

- m0A2T - Movimento do bloco A para cima da mesa.
- m0A2B - Movimento do bloco A para cima de B.
- m0A2C - Movimento do bloco A para cima de C.
- m1A2T - Movimento do bloco A para cima da mesa.
- m1C2T - Movimento do bloco C para cima da mesa.
- m0B2A - Movimento do bloco B para cima de A.
- m0C2A - Movimento do bloco C para cima de A.

- m0B2T - Movimento do bloco B para cima da mesa.
- m0C2T - Movimento do bloco C para cima da mesa.
- m0B2C - Movimento do bloco B para cima de C.
- m0C2B - Movimento do bloco C para cima de B.
- m1B2T - Movimento do bloco B para cima da mesa.
- m2B2C - Movimento do bloco B para cima de C.
- m3B2T - Movimento do bloco B para cima da mesa.
- m1C2A - Movimento do bloco C para cima de A.
- m1C2B - Movimento do bloco C para cima de B.
- m2A2C - Movimento do bloco A para cima de C.
- m3A2T - Movimento do bloco A para cima da mesa.
- m2A2B - Movimento do bloco A para cima de B.
- m2A2T - Movimento do bloco A para cima da mesa.
- m1B2C - Movimento do bloco B para cima de C.
- m1B2A - Movimento do bloco B para cima de A.
- m3C2T - Movimento do bloco C para cima da mesa.
- m2C2B - Movimento do bloco C para cima de B.
- m2C2T - Movimento do bloco C para cima da mesa.
- m2C2A - Movimento do bloco C para cima de A.
- m1A2C - Movimento do bloco A para cima de C.
- m1A2B - Movimento do bloco A para cima de B.
- m2B2T - Movimento do bloco B para cima da mesa.
- m2B2A - Movimento do bloco B para cima de A.

7 APÊNDICE B – ELEMENTOS DO MODELO PNRD

Estados:

AB - Bloco A sobre o bloco B

AC - Bloco A sobre o bloco C

AT - Bloco A sobre a mesa

Transições:

TA1 - Movimento do bloco A para cima de C

TA2 - Movimento do bloco A para cima da mesa

TA3 - Movimento do bloco A para cima de B

TA4 - Movimento do bloco A para cima da mesa

TA5 - Movimento do bloco A para cima de B

TA6 - Movimento do bloco A para cima de C