



APLICAÇÃO DO RASPBERRY PI NO PROJETO DE UM SISTEMA AUTOMATIZADO PARA ESTOCAGEM INDUSTRIAL

Pedro Augusto A. de Sousa*¹, Warlon L. N. Rosa¹, Antônio Manoel B. da Silva¹, Lúcio Rogério Júnior¹

¹UNIUBE – Universidade de Uberaba – Campus Aeroporto Uberaba

Resumo - Com o avanço da tecnologia faz-se necessário processos industriais que funcionem de forma automática, sendo capazes de controlar, possibilitar o monitoramento e armazenar informações de todos os níveis do processo. Estes sistemas são normalmente compostos por controladores lógico programáveis (CLPs), sistemas de supervisão e de aquisição de dados (SCADA) e bancos de dados. A chegada da indústria 4.0 tem exigido cada vez mais o uso dessas tecnologias, contudo, uma grande parte dos sistemas existente utiliza controladores antigos e protocolos de comunicação defasados. Com o avanço dos protocolos de comunicação e a crescente evolução dos sistemas computacionais, em alguns casos, pode-se fazer a substituição do CLP por um sistema embarcado. Com base nesse contexto, a proposta deste trabalho consiste na automação de um armazém, utilizando o Raspberry Pi, que realiza o controle do processo e a coleta dos dados que são transmitidos para um computador por meio de uma rede industrial. Para a validação do projeto, foi realizada a simulação por meio de um software 3D. O sistema proposto inclui toda a arquitetura de automação comumente encontrada no ambiente industrial atual.

Palavras-Chave – Automação industrial, Codesys, Indústria 4.0, SCADA.

APPLICATION OF RASPBERRY PI IN THE DESIGN OF AN AUTOMATED SYSTEM FOR INDUSTRIAL STORAGE

*pedrosousa@edu.uniube.br

Abstract- With the advancement of technology, it is necessary industrial processes that work automatically, being able to control, enable monitoring and store information from all levels of the process. These systems

are normally composed of Programmable Logic Controllers (PLCs), Supervisory Control and Data Acquisition (SCADA). The arrival of Industry 4.0 has increasingly demanded the use of these technologies, however, a large part of the existing systems uses old controllers and outdated communication protocols. With the advancement of communication protocols and the growing evolution of computer systems, in some cases, it is possible to replace the PLC with an embedded system. Based on this context, the purpose of this work is the automation of a warehouse, using the Raspberry Pi, which performs process control and data collection that is transmitted to a computer through an industrial network. For the validation of the project, the simulation was performed using a 3D software. The proposed system includes all the automation architecture commonly found in the current industrial environment.

Keywords - Industrial automation, Codesys, Industry 4.0, SCADA.

I. INTRODUÇÃO

O setor de logística vem passando por grandes mudanças de modo que hoje a automação é cada vez mais importante para o setor. Isto se aplica também à expedição e armazenagem, em que o aumento da qualidade e redução de custos são necessários para sobrevivência da empresa [1]. Considerando a importância do estoque para um negócio, seja ele fonte de lucro ou prejuízo, sua operação e controle devem enfatizar a gestão de estoque como estratégia empresarial [2].

Atualmente no Brasil, as indústrias estão começando a entrar em uma nova era tecnológica, a revolução industrial provocada pela indústria 4.0. Nesse cenário, observa-se que essas empresas necessitam de um elevado grau de automação. Devido à complexidade desses processos, o setor industrial aplicará uma grande soma de recursos em técnicas de automação e controle, buscando melhorar sua rentabilidade e consequentemente sua sobrevivência no mercado. É fato que

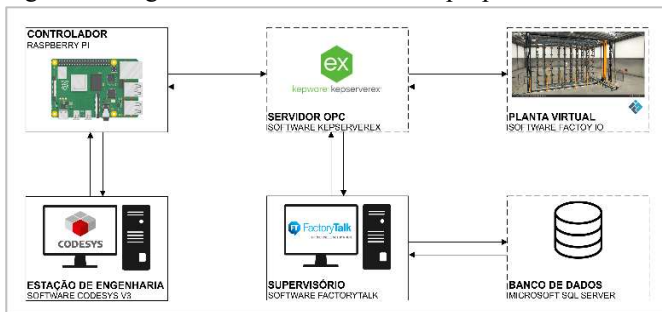
existe uma grande necessidade de aumento na produtividade do setor industrial nacional e isso demanda grandes esforços, dessa forma, a inserção de novas tecnologias tem promovido uma melhora significativa dos índices de competitividade das empresas nacionais, frente aos mercados internacionais [3].

Alinhado a este cenário, o presente trabalho apresenta uma proposta de automação de um sistema de estocagem industrial, contemplando a construção do banco de dados, de um sistema supervisorio, e a elaboração de uma lógica de controle para armazenagem e a expedição de estoque, por meio de um microcomputador de baixo custo, o Raspberry Pi.

II. MATERIAIS E MÉTODOS

Nesta seção, são apresentados os aspectos da planta virtual a ser automatizada, a topologia de rede adotada, o hardware e o software utilizados, a lógica de controle desenvolvida e a implementação por meio de uma plataforma aberta de comunicação (OPC). O diagrama de blocos abrange todos os processos de comunicação e troca de dados gerados pelos softwares e hardwares utilizados no processo (Figura 1).

Figura 1: Diagrama de blocos do sistema proposto



a. Sistema de logística e armazenagem

A logística visa transportar bens e serviços para onde são necessários dentro do prazo exigido, o que é essencial para realização das atividades de uma organização [4].

Além do transporte envolvido na logística, tem-se a expedição e a armazenagem. Neste contexto, um atraso na expedição, provoca um retardo em toda a logística de entrega e transporte de bens e serviços, assim como um atraso na armazenagem pode influenciar de forma negativa no processo industrial.

b. Factory I/O – Armazém Automatizado

Para a representação do sistema de armazenagem foi utilizada uma ferramenta de software Factory I/O, responsável por simular um processo industrial em três dimensões (3D), permitindo a construção de uma fábrica virtual, por intermédio de um cenário de uma aplicação típica de indústria.

Para o desenvolvimento do projeto, o primeiro passo foi a escolha de uma cena presente no software. A cena escolhida foi a de um setor de armazenagem. Na Figura 2, é apresentada a imagem do processo em 3D, em que estão presentes cinquenta e quatro posições para armazenagem ou expedição

do armazém, sete sensores, três atuadores, um robô para movimentação de material a ser estocado/expedido e quatro esteiras transportadoras.

Figura 2: Armazém 3D repres. o processo a ser automatizado



Na Tabela 1 são expostos os nomes das variáveis que foram utilizadas no projeto para o desenvolvimento da conexão OPC, entre Factory I/O, Raspberry e o sistema de supervisão, assim como a sua utilização na lógica de controle.

Tabela 1: Descrição das variáveis utilizadas no FactoryIO

Tipo	Nome da Variável	Descrição
Booleana	At Entry	Sensor de caixa (primeira esteira de armazenagem)
Booleana	At Load	Sensor de caixa (segunda esteira de armazenagem)
Booleana	At Unload	Sensor de caixa (primeira esteira de expedição)
Booleana	At Exit	Sensor de caixa (segunda esteira de expedição)
Booleana	At Middle	Sensor da garra do robô posição inicial
Booleana	At Left	Sensor de avanço da garra do robô (esquerda)
Booleana	At Right	Sensor de avanço da garra do robô (direita)
Booleana	Moving X	Sensor de movimento robô no eixo x
Booleana	Moving Z	Sensor de movimento robô no eixo z
Booleana	Entry Conveyor	Acionamento (primeira esteira de armazenagem)
Booleana	Load Conveyor	Acionamento (segunda esteira de armazenagem)
Booleana	Unload Conveyor	Acionamento (primeira esteira de expedição)
Booleana	Exit Conveyor	Acionamento (segunda esteira de expedição)
Booleana	Lift	Acionamento da garra do robô (sentido superior)
Booleana	Forks Left	Acionamento da garra do robô (sentido a esquerda)
Booleana	Forks Right	Acionamento da garra do robô (sentido a direita)
Inteiro	Target Position	Variável para movimentação do robô

c. Raspberry Pi – Controlador Eletrônico

O Raspberry Pi 3B+ possui uma GPIO (Pinos gerais de entradas e saídas) com 40 pinos, podendo ser usados como entradas e saídas. O Raspberry Pi 3 é mostrado na Figura 3, trata-se de um microcomputador no qual permite a instalação

de diferentes sistemas operacionais, sendo composto por 40 pinos GPIO, slot para cartão SD e 4 portas USB, bem como pinos específicos de fornecimento de tensão, conexão Ethernet integrada, WIFI e um baixo consumo de energia [5].

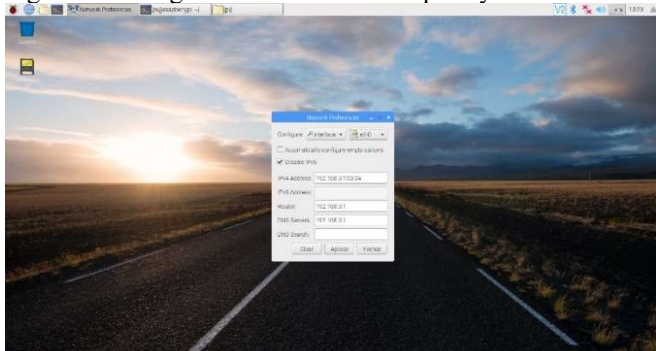
Figura 3: Raspberry Pi 3 B+ (Hardware controlador)



A escolha para utilização do Raspberry Pi foi pela sua integração com o software *CODESYS*, no qual, consta uma biblioteca denominada *Codesys Control for Raspberry Pi SL*, possibilitando a substituição de um CLP pelo hardware embarcado.

O sistema operacional *Raspberry Pi OS* foi instalado no hardware de controle, e com ele foi possível inserir o IP fixo *192.168.0.100* para estabelecer a comunicação com o computador (Figura 4).

Figura 4: Configurando IP Fixo no Raspberry PI OS



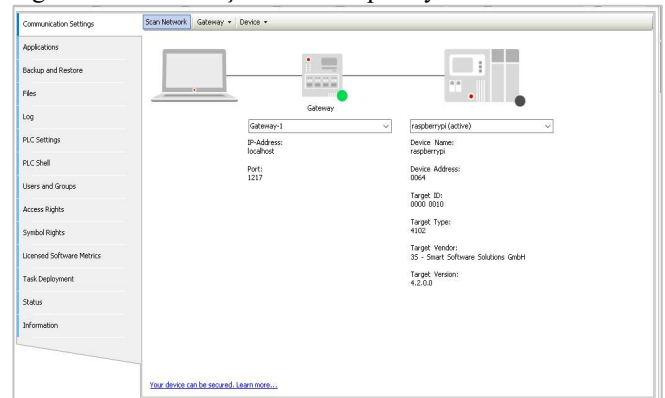
O *CODESYS* é um software desenvolvido pela empresa alemã *3S-Smart Software Solutions GmbH*, sendo uma ferramenta open-source (código aberto) e totalmente gratuita, em que muitos fabricantes de CLPs (ABB, WAGO, etc.) utilizam-no como ferramenta, para ser o ambiente de desenvolvimento de programação.

O software *CODESYS* foi instalado em um computador de uso geral, sobre *Windows 10*. A integração com o software *CODESYS* possibilitou a automação do processo utilizando de linguagens normalizada pela IEC 61131-3, como o *LD (Diagrama Ladder)* e *SFC (Diagrama Funcional Sequencial)*.

Na etapa posterior a lógica de programação para controle do sistema de estocagem foi desenvolvida. O primeiro passo foi realizar a configuração da comunicação com *Raspberry*, (Figura 5). Para realizar esta configuração, foi definido um IP fixo dentro da mesma faixa para cada equipamento, e posteriormente foi executado o comando *Scan Network* dentro do software, localizando o *Raspberry*.

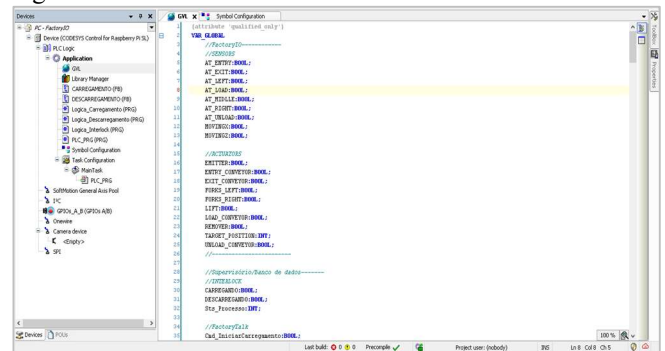
Disponibilidade de *TAGs* (nome de variáveis utilizadas pelo software) para o servidor *OPC* é realizada adicionando um objeto identificado como *Global Variable List (GVL)*.

Figura 5: Comunicação entre Raspberry e o CODESYS



As *TAGs* criadas foram demonstradas na Tabela 1, e são utilizadas para troca de informação com a planta simulada (Factory I/O), como demonstrado na Figura 6.

Figura 6: Variável Global criada no CODESYS



Para realizar a disponibilização do objeto *GVL* para o servidor *OPC* foi necessário realizar alguns passos. Acessando a guia *Symbol Configuration*, foi selecionado o objeto *GVL* como todas suas variáveis listada, e posteriormente executado o comando *Build*, para construir e compilar as informações, e por fim realizar o *Download (Comando para descarregar lógica)* para o *Raspberry PI*.

d. *KEPServerEX Protocolo de comunicação OPC UA*

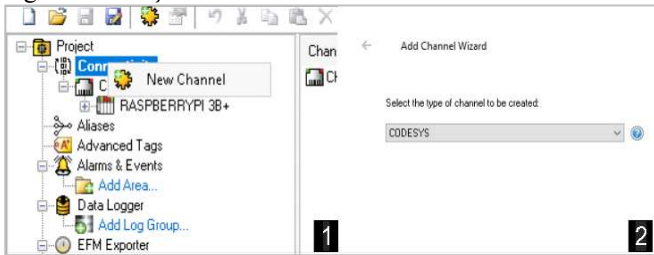
OPC UA (Open Platform Communications – Unified Architecture), é um protocolo aberto de comunicação *Ethernet*, que está numa crescente implementação nas indústrias, com seus mecanismos integrados no desenvolvimento de aplicações *IoT (Internet of Things)*, segurança integrada, independente de fornecedor ou plataforma para desenvolvimentos de aplicações.

O *KEPServerEX* é um software de servidor *OPC* com suporte de comunicação com os mais diversos equipamentos de chão de fábrica. A transferência de dados com vários sistemas supervisórios foi conseguida por meio de uma interface de comunicação cliente-servidor.

O servidor *OPC* opera como a ponte entre as informações enviadas e recebidas. A conexão é efetuada diretamente com

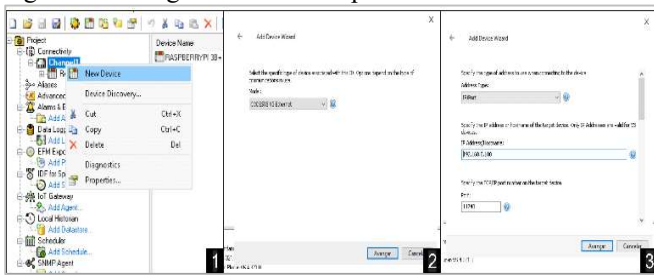
o Raspberry Pi, com a utilização da biblioteca CODESYS. Esta conexão é demonstrada a seguir (Figura 8). O primeiro passo é a criação de um canal e a escolha da biblioteca no software CODESYS, como demonstrado na Figura 7.

Figura 7: Criação de Channel KEPServerEX



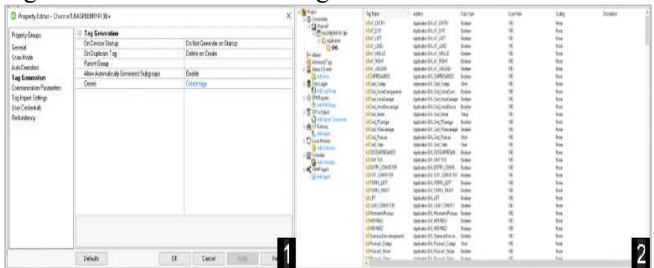
A biblioteca do CODESYS é instalada no momento da instalação do KERPServerEX. Com o novo canal criado, um dispositivo foi adicionado, nomeado de RASPBERRYPI 3B+, a versão do CODESYS foi especificada utilizado no projeto (V3 Ethernet). Assim é realizada a criação e a comunicação do novo dispositivo. Esses passos estão exibidos na Figura 8.

Figura 8: Configurando novo dispositivo



Em seguida foi necessário criar as variáveis para poder fazer o link com as variáveis disponibilizadas pelo objeto GVL. Esta etapa pode ser realizada de duas formas, em que se pode criar variável por variável e realizar o link ou gerar automaticamente essas TAGs. Esta ferramenta é denominada *Tag Generation*, sendo possível localizar o objeto GVL, selecionar as variáveis desejáveis e realizar a criação.

Figura 9: KEPServerEX – Tag Generation

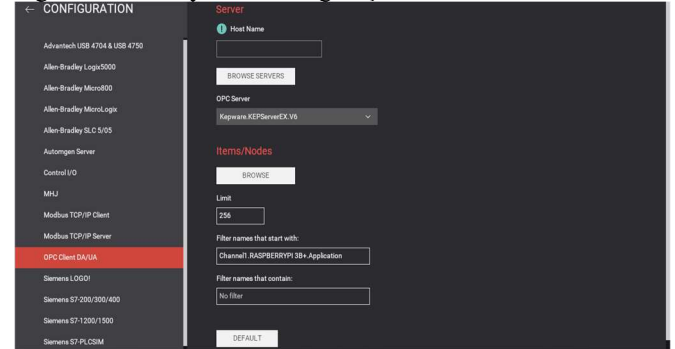


A comunicação com o software Factory I/O, é feita sucessivamente com geração das TAGs. O próprio software fornece um drive de comunicação denominado *OPC Client*

DA/UA. A Figura 10, demonstra a configuração realizada para comunicação do driver OPC Factory I/O com KEPServerEX.

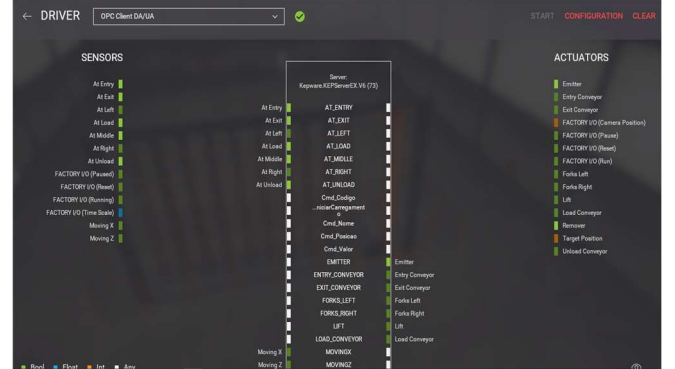
Após a criação e configuração do dispositivo no servidor OPC, é necessário fazer a busca pelo servidor realizando o comando *Browse Servers*, e com servidor localizado, um filtro de pesquisa com caminho das TAGs do objeto *GVL* foi inserido. Com esse filtro foi possível listar apenas as TAGs criadas anteriormente. Posteriormente o comando *Browse* executado para aplicar a nova configuração.

Figura 10: FactoryIO – Configuração Server



Como as TAGs criadas para comunicação com a planta simulada foram idênticas ao qual o Factory IO nomeia sensores e atuadores, apenas a conexão de TAGs com o mesmo nome utilizado foi realizada (Figura 11).

Figura 11: Factory I/O – Identificação de tags



e. Desenvolvimento Lógica de Controle

As linguagens Ladder e SFC foram usadas para desenvolvimento da lógica, conforme padronização pela IEC 61131-3. Como tem-se uma lógica sequencial e com grande repetibilidade por termos cinquenta e quatro posição para armazenagem e expedição, foi desenvolvido dois blocos funcionais, um para cada condição. Os blocos foram desenvolvidos utilizando a linguagem SFC, nomeada por alguns fabricantes de CLP como lógica *Grafcet*. Este tipo de programação consiste em um Step (passo) realizando ação, quando ele estiver acionado, e uma transição para o próximo Step (Figuras 12 e 13).

Figura 12: Lógica em Blocos para Armazenagem

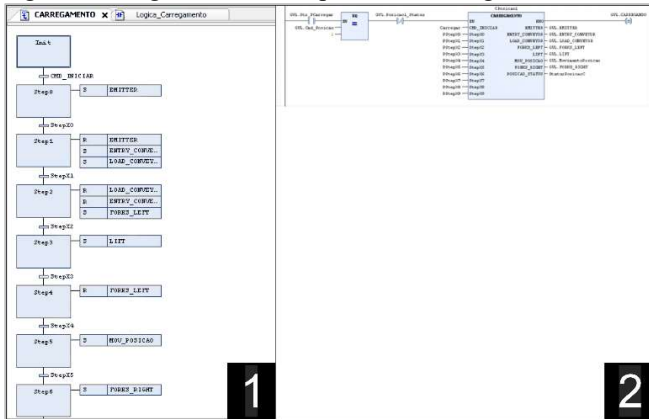


Figura 13: Lógica em Blocos para Expedição



As lógicas apresentadas nas Figura 12 e 13 foram as únicas criadas em SFC, sendo a restante desenvolvida em linguagem Ladder. Os blocos possuem semelhanças, porém para cada um deles tem-se passos com ações diferentes na planta simulada, e conseqüentemente para cada transição de Step.

Arquitetura da lógica foi dividida em quatro POU (Unidades de Organização de Programas), nos quais são:

- PLC_PRG (Rotina inicial);
- Logica_interlock (Realização de interlock entre os sistemas de armazenagem e expedição);
- Logica_Carregamento (Lógica para sistema de armazenagem);
- Logica_Descarregamento (Lógica para sistema de expedição).

POU, são objetos nos quais realizam a integração de programas, blocos funcionais e funções. A junção desses elementos permite que a programação seja realizada em nível de objeto Local ou Global.

PLC_PRG, está rotina principal, e realiza o comando de liberar as outras rotinas, demonstrado pela Figura 14.

Figura 14: POU – PLC_PRG



A Lógica de carregamento/armazenagem ou descarregamento/expedição, é bastante similar, contendo estruturação com verificações para início de processo, condições para avanço de Step e bloco com condições para cada posição da prateleira.

As condições de avanço de cada Step foram realizadas em Ladder, pela simplicidade da linguagem gráfica e facilidade de um avanço com várias condições, possibilitando-se fazer análises e detecção de erros de forma fácil e interativa. Nas Figuras 15 e 16 são apresentadas a lógica da Armazenagem e Expedição.

Figura 15: Linha de código p/ controle da Armazenagem

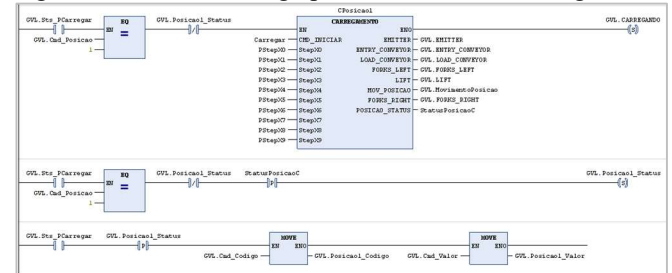
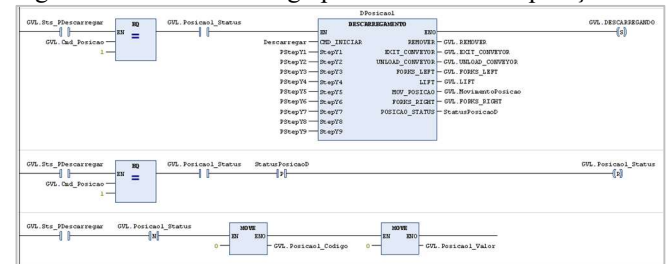


Figura 16: Linhas de código para controle da Expedição



Por meio das Figuras 15 e 16, pode-se observar a atuação da lógica para cada posição, em que são considerados um total de 54 conjuntos, um para cada posição.

Os comandos de posição, código e valor, são inseridos pelo sistema supervisor e posteriormente disponibilizados para o banco de dados. Cada posição possui suas respectivas TAGs, seguindo um padrão e mudando o valor final com a posição referida. Quando houver uma expedição, os valores das variáveis serão zerados, indicando que não contém nenhum material estocado naquela posição.

Na lógica tem-se um interlock de segurança para não ocorrer o carregamento de material na mesma posição, bem como deixar executar o processo de expedição numa posição vazia, evitando custo adicionais no processo. Outro ponto é a presença de um interlock envolvido no momento da execução dos processos, no qual pode ser descrito como um intertravamento entre os processos, evitando uma paralização no processo por erro de comando, é liberado quando finalizado.

III. RESULTADOS

Os resultados demonstrados a seguir, referem-se à comunicação com Raspberry Pi, comunicação das variáveis

com o servidor OPC é comissionamento da lógica desenvolvida.

Teste de comunicação com Raspberry Pi

Nesta etapa foi realizado o teste para verificação do IP fixo definido (Figura 17). O teste foi realizado utilizando-se o comando PING no Prompt de Comando do Windows.

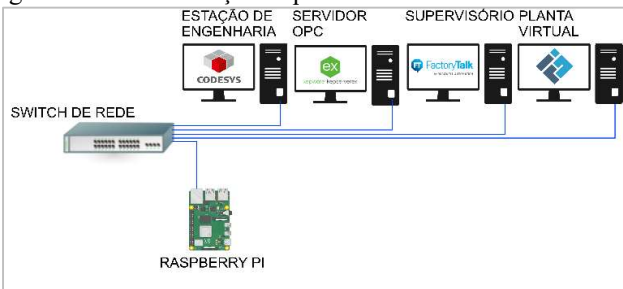
Figura 17: Teste de comunicação - Ping com Raspberry Pi

```
C:\Users\W\In10>ping 192.168.0.100
Disparando 192.168.0.100 com 32 bytes de dados:
Resposta de 192.168.0.100: bytes=32 tempo<1ms TTL=64
Resposta de 192.168.0.100: bytes=32 tempo=1ms TTL=64
Resposta de 192.168.0.100: bytes=32 tempo=1ms TTL=64
Resposta de 192.168.0.100: bytes=32 tempo<1ms TTL=64
Resposta de 192.168.0.100: bytes=32 tempo<1ms TTL=64

Estadísticas do Ping para 192.168.0.100:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
                perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 0ms, Máximo = 1ms, Média = 0ms
C:\Users\W\In10>
```

A arquitetura de rede construída, consiste em todos os dispositivos estarem na mesma infraestrutura, e com as mesmas faixas de IP (Figura 18).

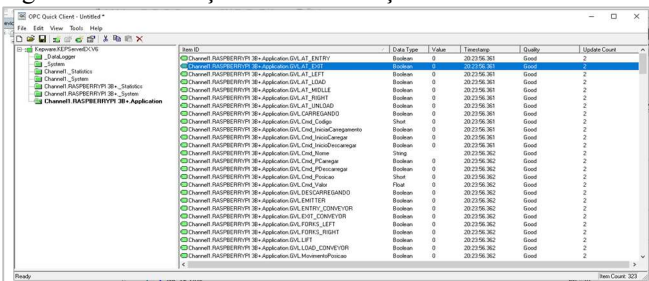
Figura 18: Demonstração arquitetura de rede



Comunicação das variáveis do servidor OPC

Para avaliação da comunicação das variáveis do servidor OPC com o Raspberry Pi e o sistema de supervisor, é realizado o teste pelo software KEPServerEX, por intermédio da ferramenta OPC Quick Cliente, verificando a qualidade da comunicação em boa ou ruim. Conforme demonstrado pela Figura 19, obteve-se todas as comunicações de variáveis com os dispositivos da rede com um forte sinal e sem perdas.

Figura 19: Verificação da comunicação das variáveis



Verificação comunicação Factory I/O com Servidor OPC

A validação da comunicação do Factory I/O com o servidor OPC foi realizada por meio do acesso ao software de simulação, sendo feita por intermédio do acesso à configuração de drive, e verificado na aba OPC Cliente

DA/UA a constatação da comunicação efetiva com o servidor, sinalizada com a cor verde.

Comissionamento do sistema de estoque industrial

Para realização do comissionamento do sistema, e o teste de toda arquitetura de automação comumente encontrada nos ambientes industrial, foi realizado os processos de armazenagem e a expedição. Os testes envolvidos, foram para as cinquenta e quatro posições, intertravamentos e comunicação com os sistemas de supervisor e banco de dados (Figura 20).

Figura 20: Comissionamento sistema



III. CONCLUSÕES

Neste trabalho foi possível realizar a aplicação do Raspberry Pi em um sistema automático de controle de estoque industrial, demonstrando a possibilidade de utilizar uma ferramenta de baixo custo, fácil acesso comercial, extremamente difundida no meio técnico e acadêmico, realizando com êxito um controle de processo tipicamente realizado por um CLP e integrando-se a um sistema de automação comumente encontrada no ambiente industrial.

REFERÊNCIAS

- [1] CÂMARA, V. de A. et al. Proposta de automação de estoque com aplicação do dmaic em uma empresa de cosméticos: um estudo de caso, [s. l.], 2019.
- [2] ARAÚJO, B. R. M. Dispositivo implementado por IoT para contagem de produtos em gôndolas e processo de automação de estoques a distância. 2021. Trabalho de conclusão de curso (Mestre em propriedade intelectual e transf. de tecnologia para inovação) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, [S. l.], 2021.
- [3] ZANATA, D. R. P. (2005). Desenvolvimento de sensor virtual empregando redes neurais para medição da composição em uma coluna de destilação. Dissertação de Mestrado, Universidade de São Paulo.
- [4] CARVALHO, B. A. N.; GASQUES, A. C. F. Avaliação do retorno do investimento para automação de um processo de paletização. 2017. Trabalho de Conclusão de Curso (Graduação engenharia de produção) - Universidade Estadual de Maringá, [S. l.], 2017.
- [5] Datasheet Raspberry Pi Module 3+. Disponível em: https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf. Acesso em: 10 de maio de 2022.