



FACULDADE SENAI DE TECNOLOGIA MECATRÔNICA

INTERFACE E CONTROLE DE SIMULAÇÃO ENTRE ROBOGUIDE E CLP SIEMENS

SIMULATION INTERFACE AND CONTROL BETWEEN ROBOGUIDE AND PLC SIEMENS

William Gregorio da Silva^{1, i}

Júlio Cesar de Almeida Freitas^{2, ii}

Paulo Sebastião Ladivez^{3, iii}

RESUMO

O uso de *softwares* para realizar processo de simulação de um projeto de máquinas são cada vez mais utilizados na engenharia. Com o advento da indústria 4.0 é cada vez mais iterativo a necessidade da simulação para a aprovação de um projeto, acrescentado a validação do controle de automação. O uso de *softwares* para validar virtualmente a realidade entre os elementos cinemáticos e seus controles se tornam salutar para a demonstração do funcionamento das relações entre os equipamentos a tornar viável o projeto. Esta característica somada a possibilidade de planejar, alterar, reagrupar as situações, se fazem presente para uma acurácia fiel a antecipar a construção e montagem real do projeto.

ABSTRACT

The use of software to carry out the simulation process of a machine project is increasingly used in engineering. With the advent of Industry 4.0, the need for simulation to approve a project is increasingly iterative, with the addition of automation control validation. The use of software to virtually validate the reality between the kinematic elements and their controls becomes beneficial for demonstrating the functioning of the relationships between the equipment to make the project viable. This feature, added to the possibility of planning, altering, regrouping situations, are present for a faithful accuracy to anticipate the actual construction and assembly of the project.

1 INTRODUÇÃO

Com a constante evolução tecnológica, o processo de simulação e comissionamento virtual é cada vez mais necessário, a tornar cada vez mais comum a exigência na prestação de serviço para as montadoras em nosso parque fabril. Com o advento da quarta revolução industrial, conhecida como Indústria 4.0 e com a manufatura digital, o objetivo desse trabalho é demonstrar de forma objetiva e prática o processo da criação de uma interface de controle e simulação de uma estação robotizada. Desta maneira, será possível avaliar e identificar possíveis erros de trajetória da movimentação do robô e suas interferências mecânicas, assim

¹ Graduando em Tecnologia Mecatrônica Industrial pela Faculdade SENAI de Tecnologia Mecatrônica. E-mail: willian_gsilva@hotmail.com

² Professor da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: julio.freitas@sp.senai.br

³ Professor da Faculdade SENAI de Tecnologia Mecatrônica. E-mail: paulo.ladivez@sp.senai.br

como também validar todas as lógicas de controle desenvolvidas nos programas de PLC e certificar o funcionamento da interface de operação com o usuário.

Neste contexto, é buscado a redução de tempo a evitar retrabalhos e execuções desnecessárias nas tentativas de funcionamento da lógica mais eficiente. Nesta fase de desenvolvimento do projeto é eliminado todos os possíveis erros de lógicas tão quanto de trajeto para a execução da atividade em análise.

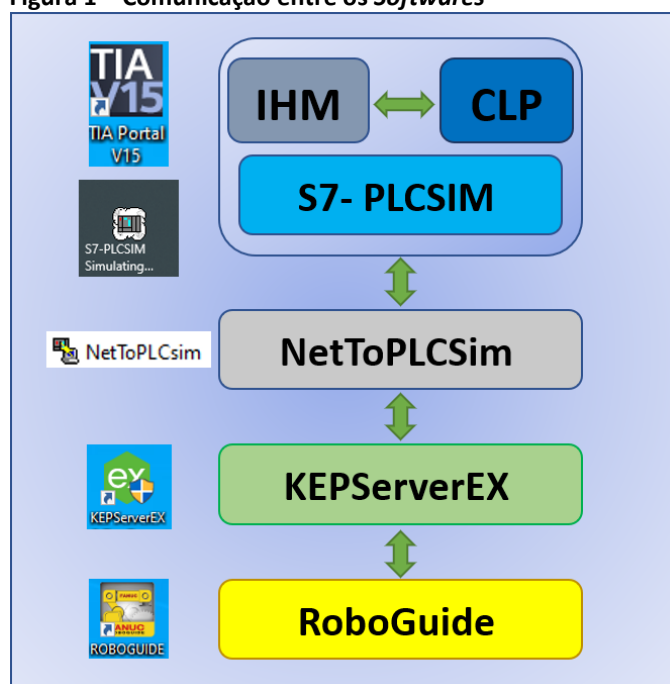
2 DESENVOLVIMENTO

Para o desenvolvimento deste projeto é utilizado quatro plataformas de *softwares*, na integração robótica, com o controlador lógico programável e a interface homem máquina de operação, conforme segue:

- a) *Roboguide V9 (Ver.S) – FANUC*
Software para criação e simulação das células robotizadas
- b) *Totally Integrated Automation PORTAL (TIA PORTAL V15) – SIEMENS*
Software para controle e interface da célula robotizada
- c) *KEPServerEX – Kepware – V6.10.623*
Software para interface de sinais de controle entre o TIA PORTAL e *Roboguide*
- d) *NetToPLCSim V1.2.4.0*
Software para simulação dos sinais de comunicação em rede do controlador

O objetivo nesta aplicação é a simulação da interação da célula robotizada com o Controlador Lógico Programável (CLP) em conjunto com a interface homem máquina (IHM). A execução da atividade na célula robotizada é elaborada, planejada no software de simulação do robô, no envio e recebimento de sinais para a execução da lógica, através de um servidor que possibilitará a interação entre os clientes. Este procedimento, é mostrado na figura 1 do gráfico de comunicação entre os softwares.

Figura 1 – Comunicação entre os Softwares



Fonte: Elaborado pelo autor

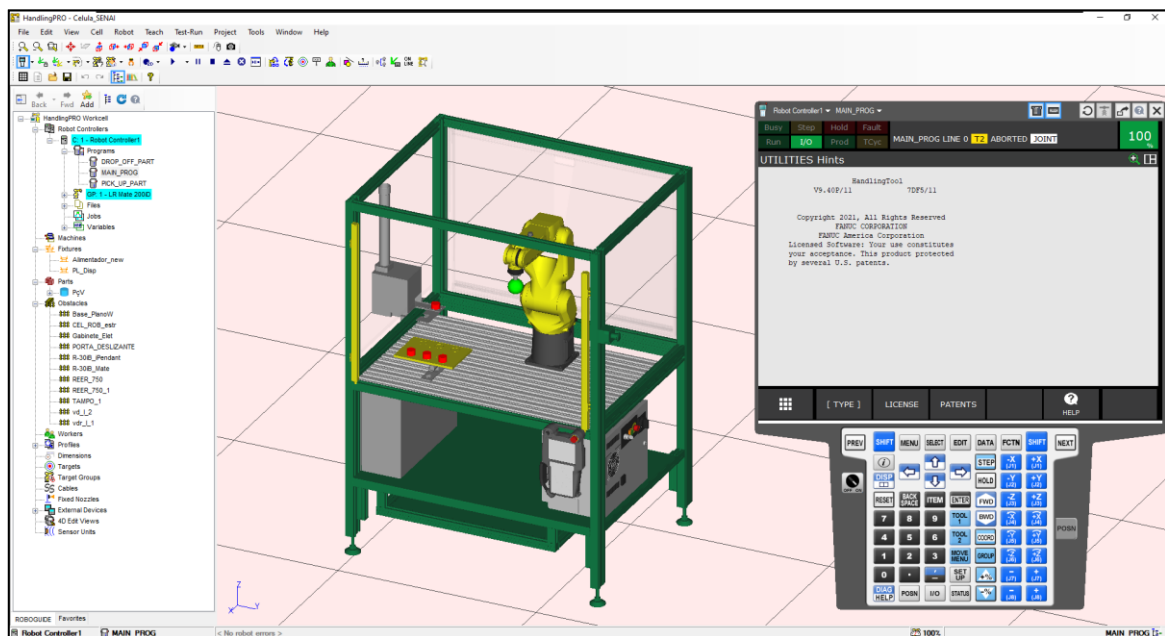
2.2 Características dos Softwares

Segue abaixo uma breve descrição dos *softwares* que foram utilizados nesse projeto de controle e simulação de uma célula robotizada.

2.2.1 Roboguide

O *Roboguide* é um *software* desenvolvido pela *FANUC* para criação e simulação de ambientes de células robotizadas. Ele possui uma grande variedade de modelos de robôs com inúmeras características para diversas aplicações. Nesse projeto utilizamos a aplicação *Handling PRO* com o robô de modelo *LR Mate 200iD* (figura 2). A programação da atividade do robô, foi concebida, elaborada e simulada no ambiente do *software Roboguide*.

Figura 2 - Roboguide

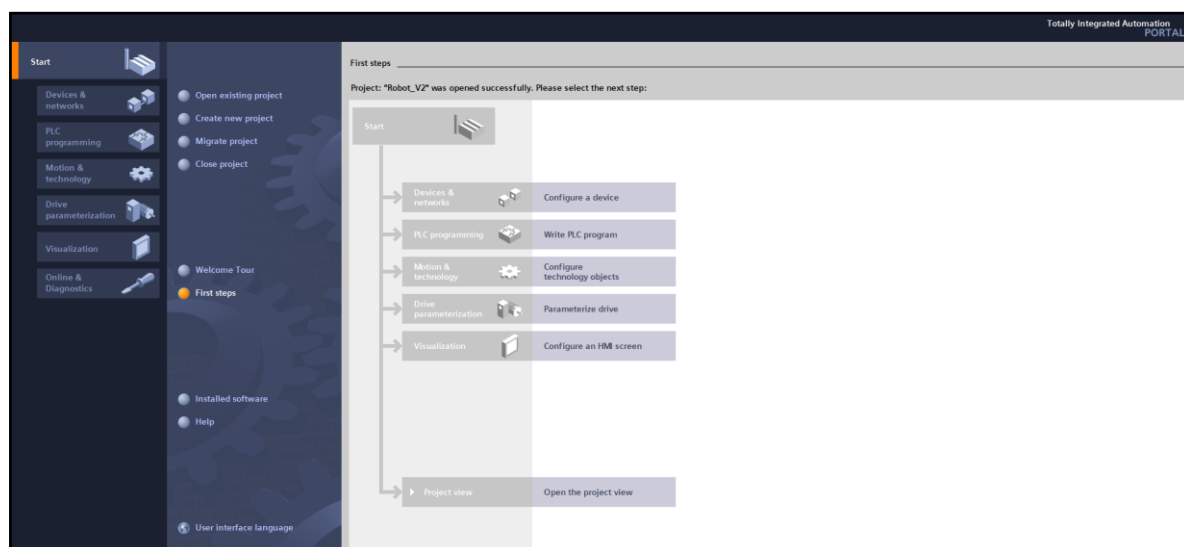


Fonte: Elaborado pelo autor.

2.2.2 TIA PORTAL

Para criação das lógicas de controle e de interface com o operador, foi utilizado o *TIA PORTAL* (figura 3) desenvolvido pela *SIEMENS*. Por ser um *software* de plataforma integrada, em um mesmo projeto foi criado as telas de interface, blocos lógicos para controle dos sinais e simulação do *hardware*, *software* e *IHM*. Como recurso já integrado e disponível no *TIA PORTAL*, também se fez uso da aplicação *S7 PLCSIM* que possibilita desenvolver um ambiente de simulação do hardware do CLP, CPU, módulos de entrada e saídas digitais, etc.

Figura 3 – TIA PORTAL

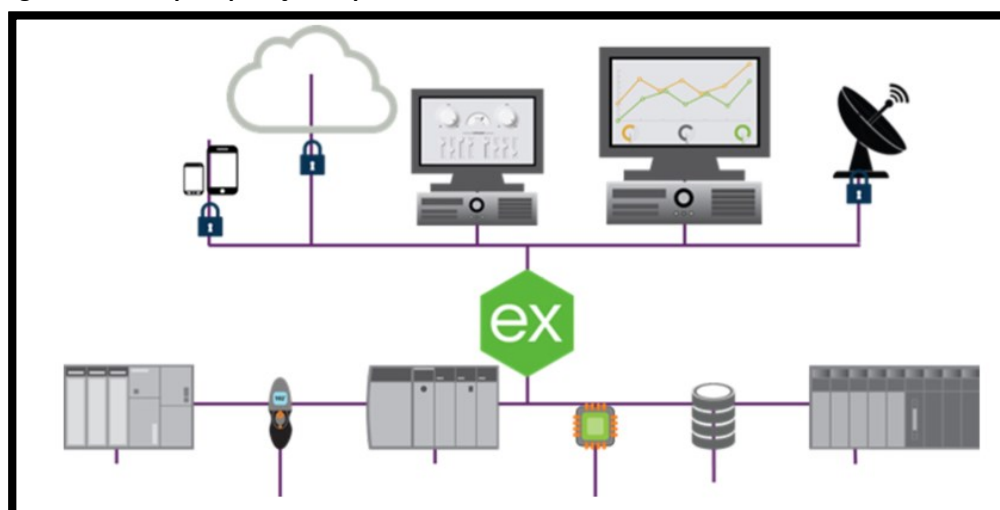


Fonte: Elaborado pelo autor.

2.2.3 KepServerEX

Este recurso é um *software*, desenvolvido pela *KEPWARE*, capaz de criar uma plataforma de comunicação industrial de acesso de dados entre o controlador do robô e o CLP através do protocolo de comunicação *OLE for Process Control (OPC)*. É através da plataforma, *KepServerEX*, pela qual possibilita-se criar elementos de interface entre os blocos lógicos de controle do CLP e direcioná-los às entradas e saídas do controlador do robô. Na figura 4 é mostrado um exemplo de aplicação desta plataforma.

Figura 4 – Exemplo aplicação KepServerEX



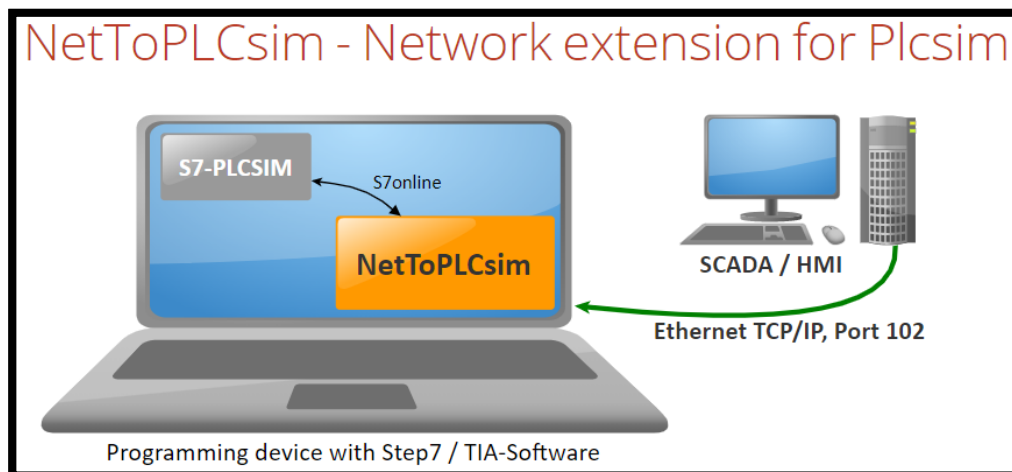
Fonte: KepWare

2.2.4 NetToPLCsim

Este recurso possibilita acessar a aplicação do S7-PLCSIM via uma rede de comunicação através do protocolo *Transmission Control Protocol* e Internet Protocol (TCP/IP). De forma mais simples, é possível definir que o *software NetToPLCSim*, é responsável por simular uma

comunicação entre a placa de rede do CLP com a placa de rede do controlador do robô, entretanto, utilizando as configurações da placa de rede do próprio computador, que realiza toda a simulação, com pode ser demonstrado na figura 5.

Figura 5 - NetToPLCSim



Fonte: NetToPLCSim Fig. 5

2.3 Tecnologia de aplicação

Para a aplicação de controle através do CLP, foi desenvolvido um programa simples de movimentação do robô na célula robotizada. Essa simulação tem como objetivo demonstrar a movimentação do robô para sujeitar algumas peças em um dispositivo. Os comandos de partida do ciclo, pausa do programa, cancelar falhas, entre outros, serão controlados através da IHM do CLP.

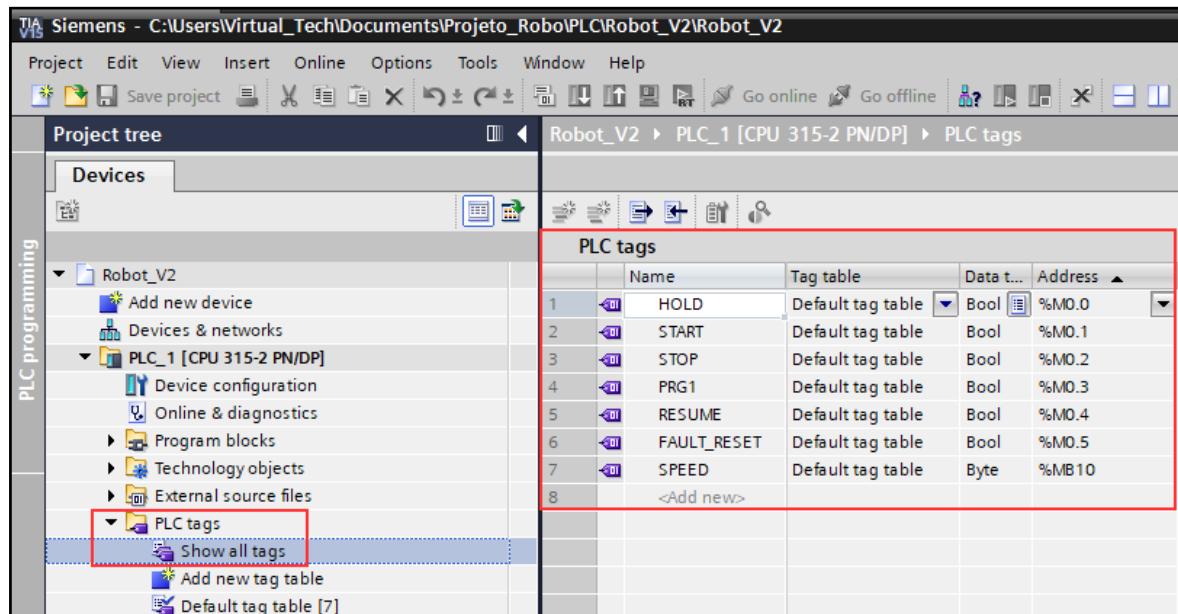
O modelo de CPU utilizada nesse projeto foi a CPU 315-2 PN/DP. É um modelo de CPU desenvolvida pela SIMENS, a qual possui uma porta de comunicação PROFINET (X2). Será através dessa porta que iremos estabelecer a interface de comunicação com o controlador do robô.

PROFINET “é uma rede baseada em um padrão de comunicação Ethernet Industrial padronizado pelas normas IEC61158-5 e IEC 61158-6”. (WIKIPEDIA 2020).

2.3.1 Interface de controle do robô (IHM)

Foram criados sinais de interface de controle do robô. Através de um elo de comunicação entre o controlador do robô e CLP, esses sinais serão utilizados na IHM para acesso do operador aos controles do robô. Para isso, foi criado algumas TAG's no programa do CLP (Figura-6).

Figura 6 – TAG's



Fonte: Elaborado pelo autor

Essas TAG's são usadas como elementos de interface entre a tela de operação e os sinais de controle do robô.

- a) *Hold (BOOL)* – Pausa o programa em andamento do robô
- b) *START (BOOL)* – Partida do programa a ser executado no robô
- c) *STOP (BOOL)* – Para o programa em andamento do robô
- d) *PRG1 (BOOL)* – Seleção do programa 1 configurado no controlador do robô
- e) *RESUME (BOOL)* – Reinício do programa em andamento no robô após uma pausa
- f) *FAULT_RESET (BOOL)* – Cancela falha do controlador do robô
- g) *SPEED (Byte)* – Configura velocidade em porcentagem (0-100) do controlador do robô

O termo *BOOL* é utilizado na ciência da computação como abreviação de elementos booleanos (*boolean*). É um tipo de dado primitivo que possui dois valores, que podem ser considerados como 0 ou 1, falso ou verdadeiro. Termo usado em homenagem ao seu criador, George Boole, que definiu um sistema de lógica algébrica pela primeira vez na metade do século XIX. (WIKIPÉDIA, 2020)

Para a interação do operador, foi criada uma tela de interface simples com as TAG's de controle. Utilizamos o modelo *TP700 CONFORT* da *SIEMENS*, porém, através do *TIA PORTAL* é possível usar toda a criatividade para elaboração de telas com diferentes tipos de design de aplicação. Na figura 7 é mostrado o exemplo de tela principal que foi criado para interface com o operador.

Figura 7 – Tela principal



Fonte: Elaborado pelo autor

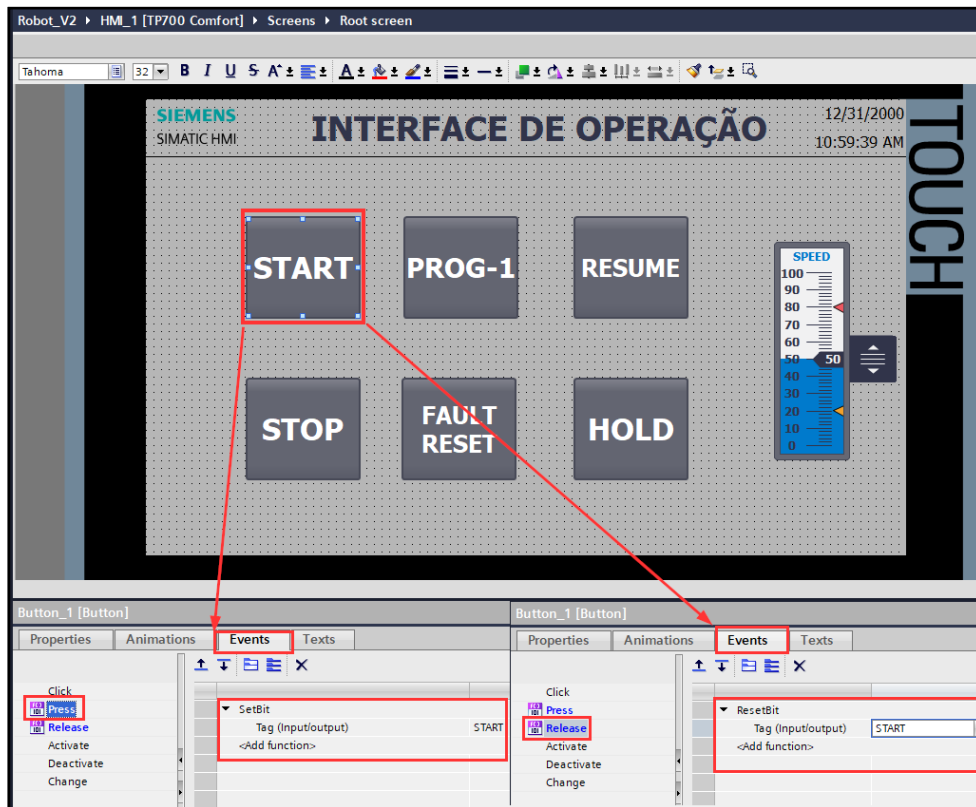
Na tela principal, foi criado seis (6) botões de operação e uma barra deslizante para determinar a velocidade de avanço do programa do robô. Para cada botão e para a barra deslizante foi configurado suas TAG's em função de cada tipo de operação. A partir da figura 8 é demonstrado o exemplo de configuração dos botões.

Para os botões *START*, *PROG-1*, *RESUME*, *STOP* e *FAULT RESET* foi atribuído a configuração *Event* do tipo *SetBit* na aplicação *Press* e *ResetBit* na aplicação *Release*.

No botão *HOLD* foi utilizado a configuração *ResetBit* na aplicação *Press* e *SetBit* na aplicação *Release*. Isso deve-se pelo fato da entrada do sinal *HOLD* no controlador do robô funcionar com estado de nível alto, ou seja, quando o botão não estiver sendo pressionado, o nível lógico dessa entrada deverá ser verdadeiro (nível lógico 1). Caso o botão *HOLD* seja pressionado, esse sinal será falso, ou seja, nível lógico 0 (figura 9).

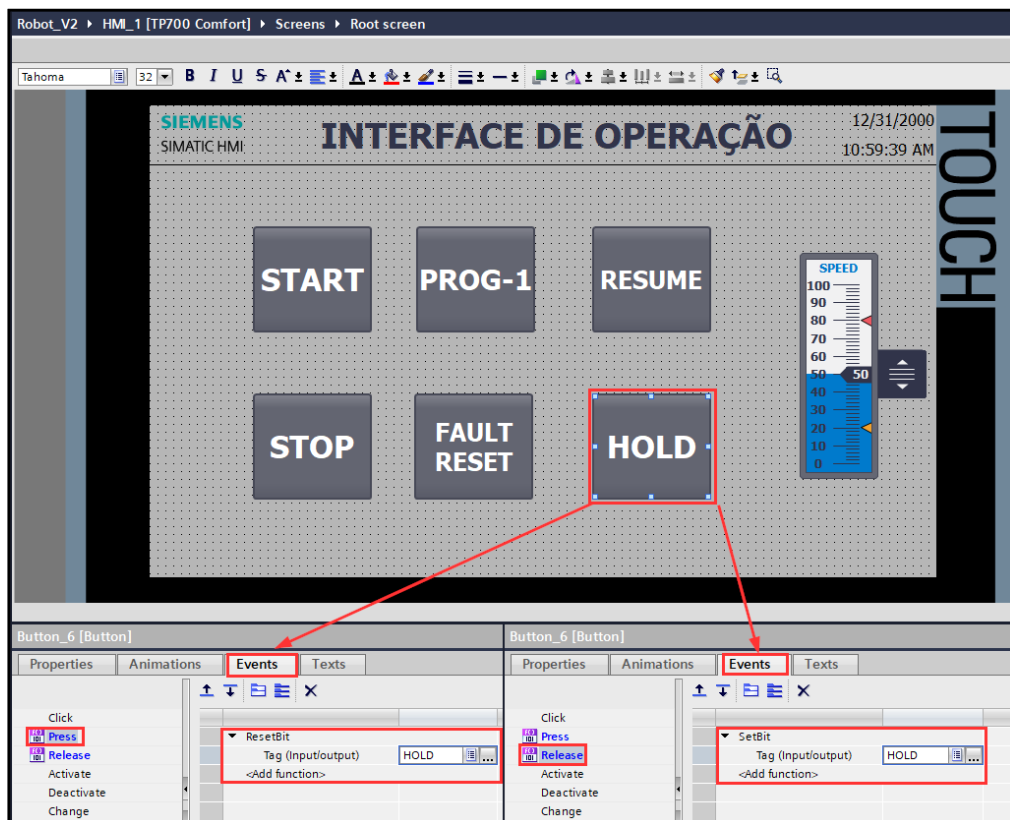
Para o controle de velocidade do avanço do robô, foi utilizado a aplicação de barra deslizante (*Slider*). Nessa aplicação foi atribuído a TAG *SPEED* e, os seus limites foram configurados com o mínimo 0 e o máximo 100. Esse valor de 0 a 100 será utilizado no controlador do robô como a taxa em porcentagem da velocidade do avanço do robô, ou seja, de 0 a 100 por cento da velocidade máxima configurada do robô (figura 10).

Figura 8 – Configuração dos botões



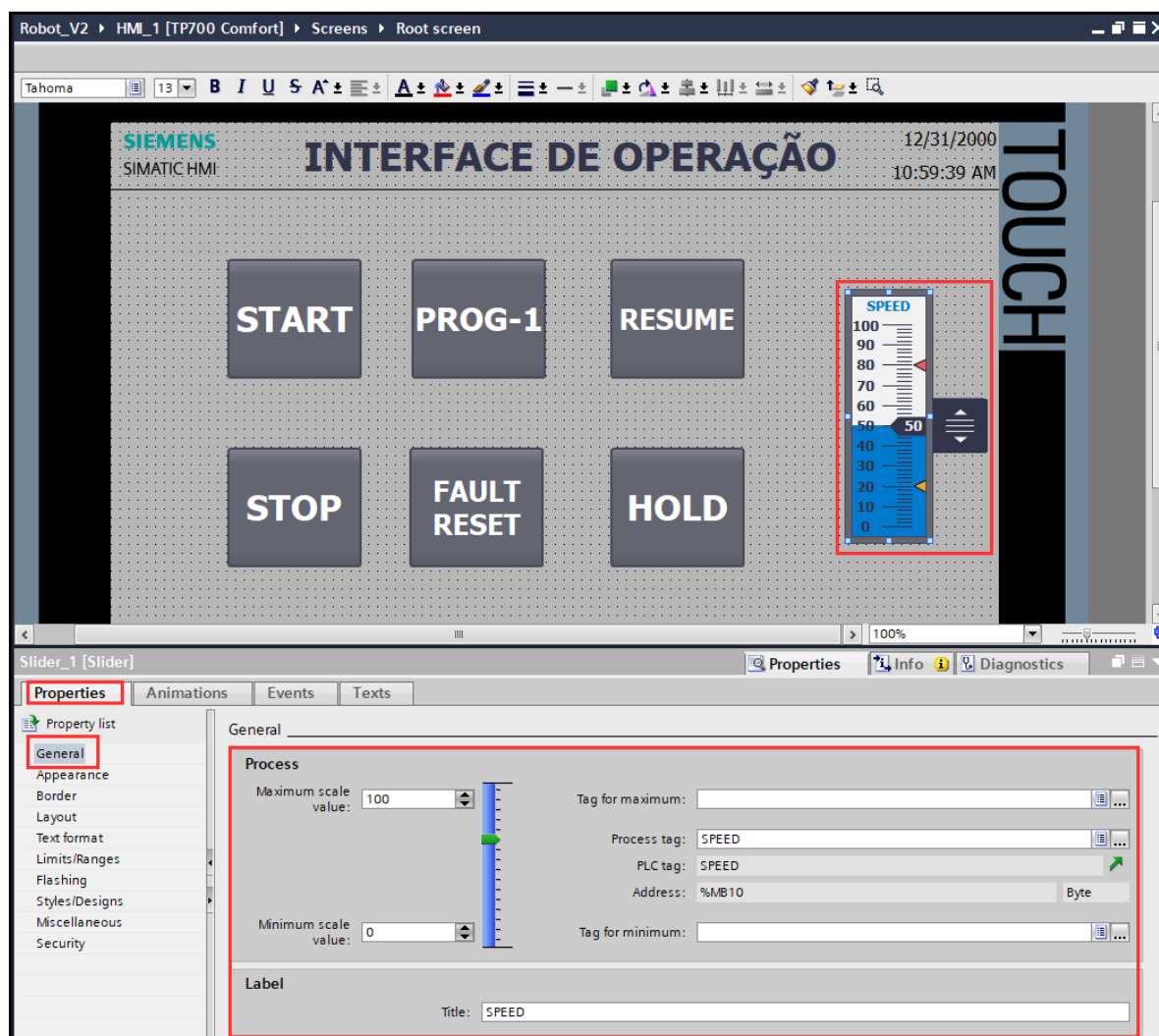
Fonte: Elaborado pelo autor

Figura 9 – Configuração botão HOLD



Fonte: Elaborado pelo autor

Figura 10 – Configuração da barra deslizante



Fonte: Elaborado pelo autor

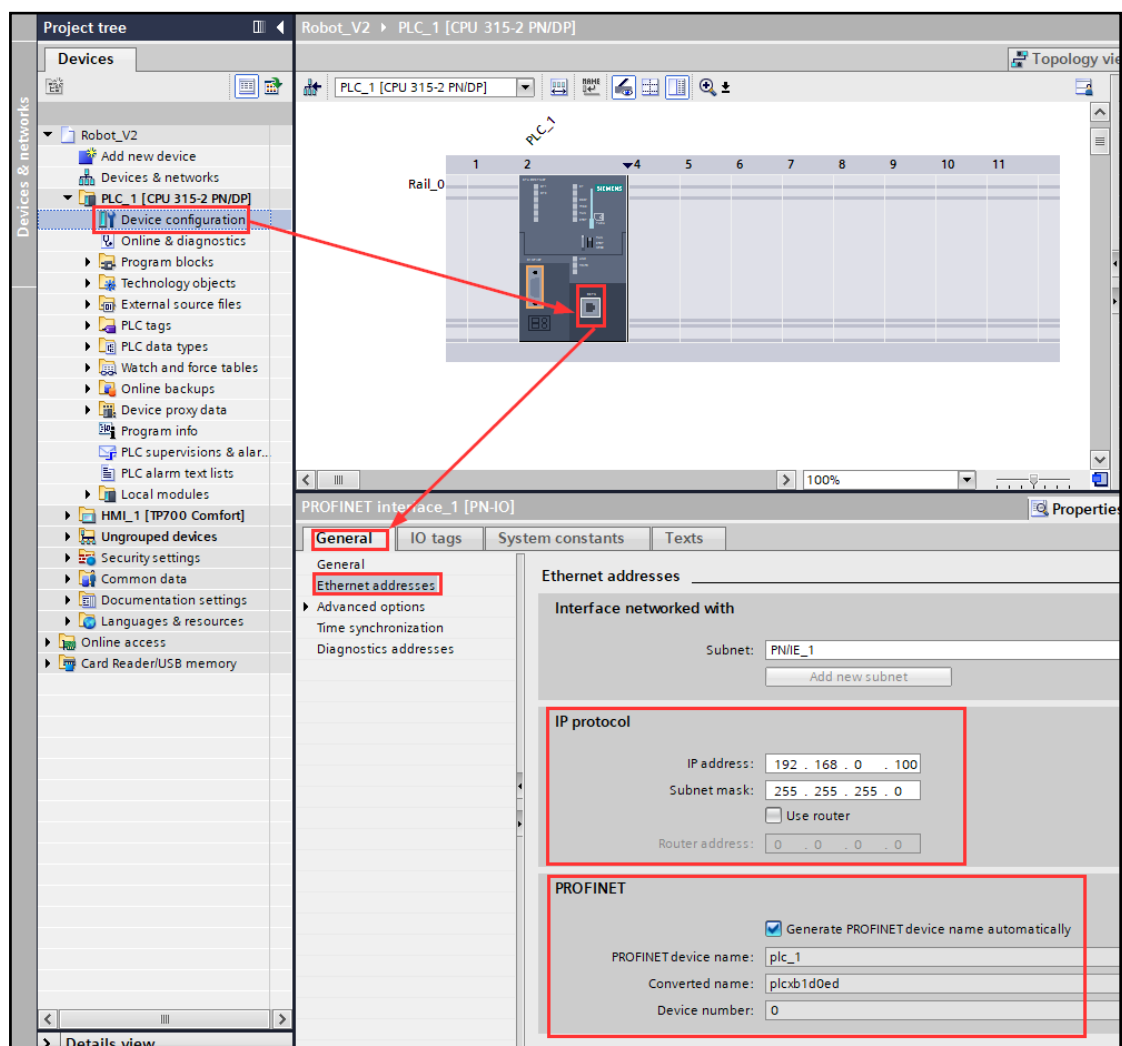
2.3.2 Controlador Lógico Programável (CLP)

Como exemplo de arquitetura de aplicação, nesse projeto foi utilizado o modelo de CPU 315-2 PN/DP da *SIMENS*. Esse modelo de CPU usa uma porta de comunicação de rede que será configurada com o protocolo Ethernet/IP, conforme demonstrado na figura 11.

Através da CPU é possível elaborar blocos lógicos de programas de controle da automação. Na arquitetura desse projeto, a CPU é utilizada como ponte de comunicação entre a IHM e controlador do robô, com objetivo de enviar os sinais de controle previamente configurados na tela de IHM.

Existem aplicações de projetos que precisam ter soluções de arquitetura de lógicas mais elaboradas para um nível de automação mais complexa. O CLP poderá ser utilizado como gerenciador de tarefas do robô, pois o mesmo poderá controlar outros periféricos de entradas e saídas de sinais em conjunto com o controlador do robô.

Figura 11 – Configuração da CPU



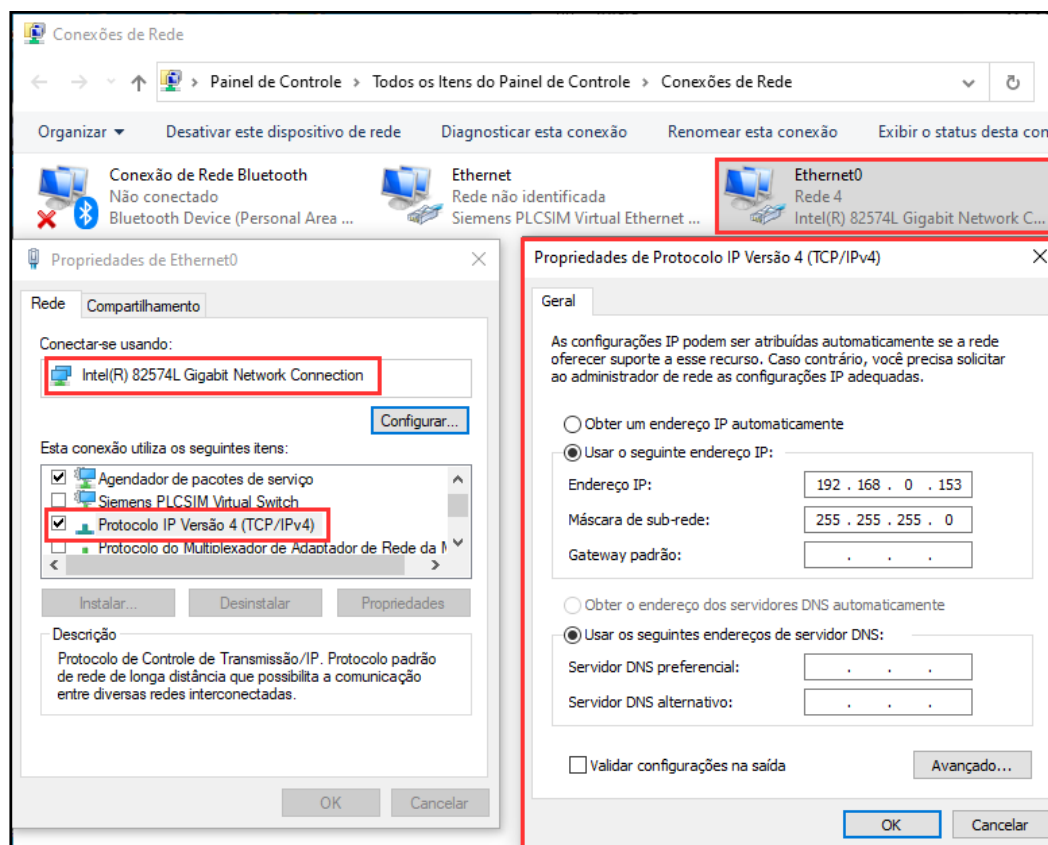
Fonte: Elaborado pelo autor

2.3.3 Interface de comunicação (KepServerEX e NetToPLCSim)

Após a criação da tela de interface do operador e da configuração da CPU, é necessário estabelecer a comunicação entre o CLP e o controlador do robô. Para isso, precisamos configurar os *softwares* KepServerEX e o NetToPLCSIM.

O protocolo de comunicação definido é o TCP/IP, portanto, é necessário configurar o adaptador de rede do computador na mesma faixa de endereço IP que foi configurado a CPU do CLP como mostra a figura 12.

Figura 12 – Configuração do adaptador de rede do computador

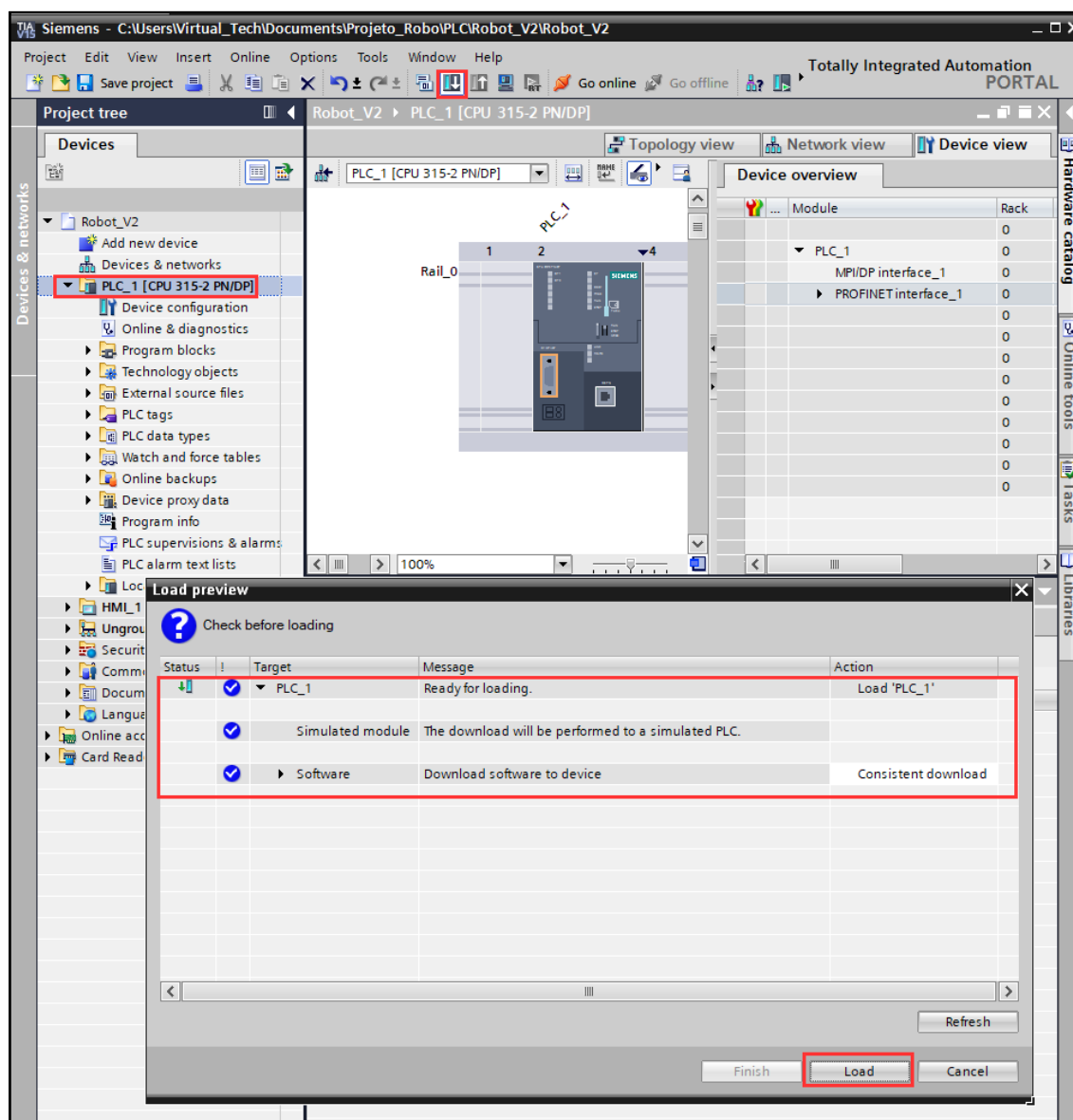


Fonte: Elaborado pelo autor

A próxima etapa é realizar as configurações de comunicação entre os *softwares* S7-PLCSIM e o NetToPLCSim.

Para esta configuração, o procedimento é executar este *software* no menu de programas do sistema operacional. É preciso compilar o projeto criado no TIA PORTAL e descarregar na CPU virtual do S7-PLCSIM, como demonstrado na figura 13.

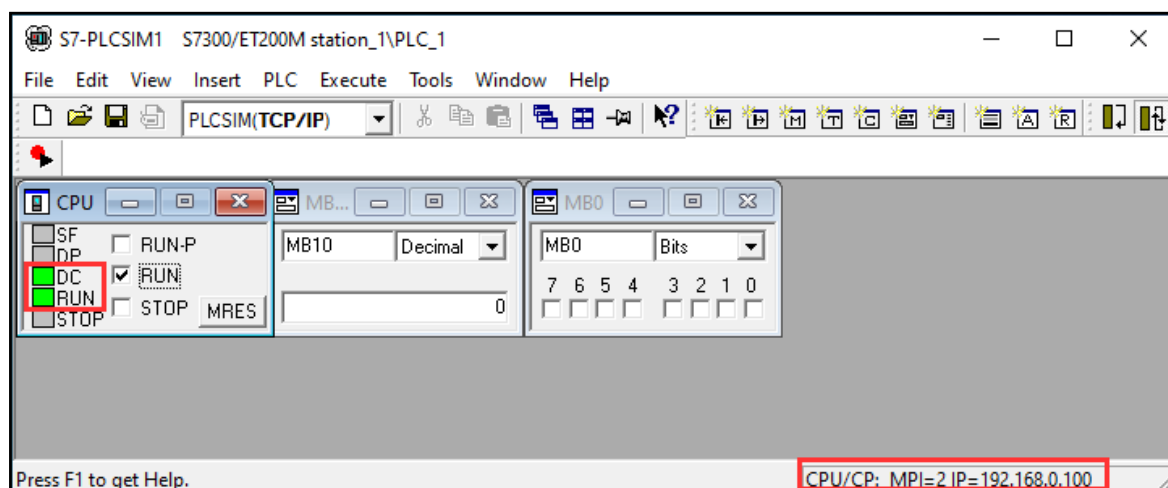
Figura 13 – Descarregar projeto do TIA PORTAL



Fonte: Elaborado pelo autor

Concluído o processo de transferência do projeto do TIA PORTAL no S7-PLCSIM, o simulador da CPU estará com as configurações de *hardware* e de *software* do projeto. Nesta fase, a CPU está configurada para entrar no estado "RUN". Na figura 14 é possível visualizar a tela de indicação da CPU no estado "RUN"

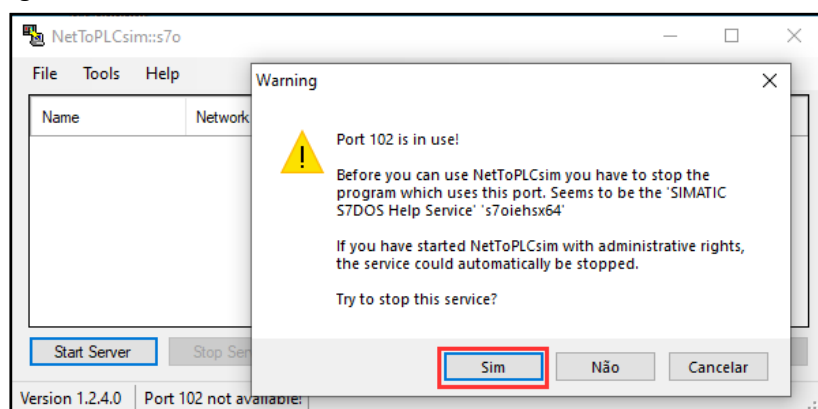
Figura 14 – S7-PLCSIM



Fonte: Elaborado pelo autor

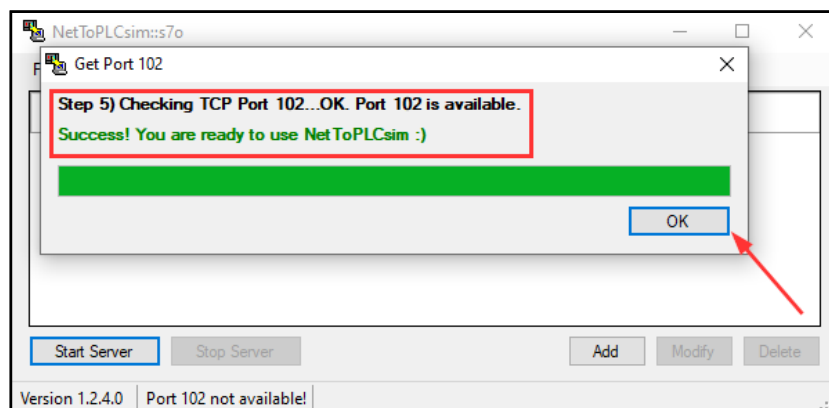
Nesta etapa, o simulador está operacional e é possível configurar o *software* NetToPLCSim, conforme as figuras 15, 16, 17, 18, 19 e 20 a seguir.

Figura 15 – NetToPLCSim



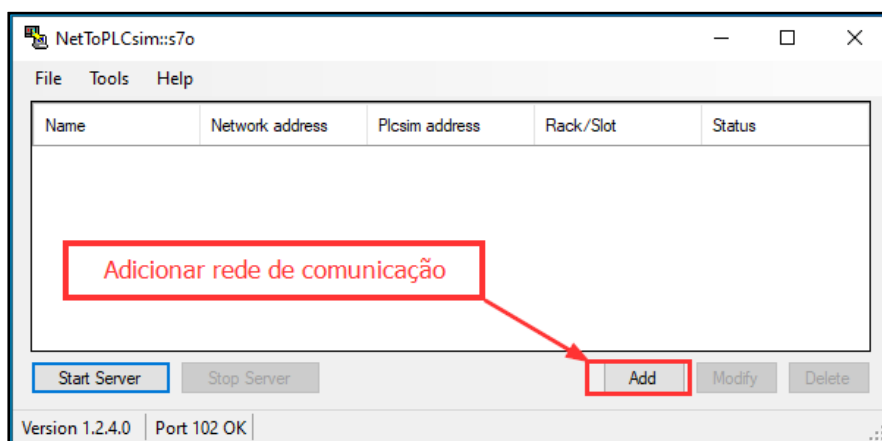
Fonte: Elaborado pelo autor

Figura 16 – NetToPLCSim



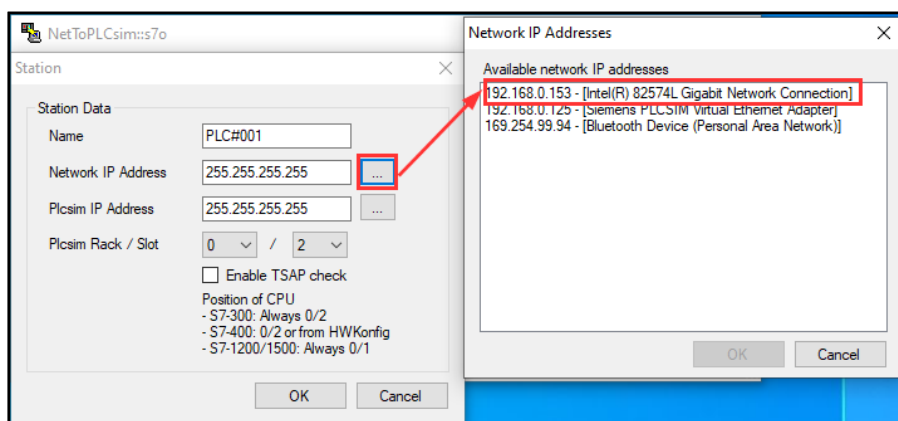
Fonte: Elaborado pelo autor

Figura 17 – NetToPLCSim



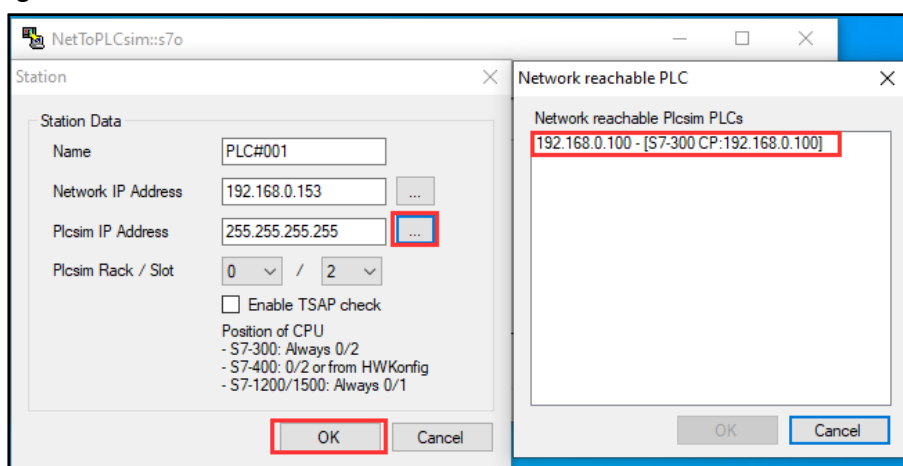
Fonte: Elaborado pelo autor

Figura 18 – NetToPLCSim



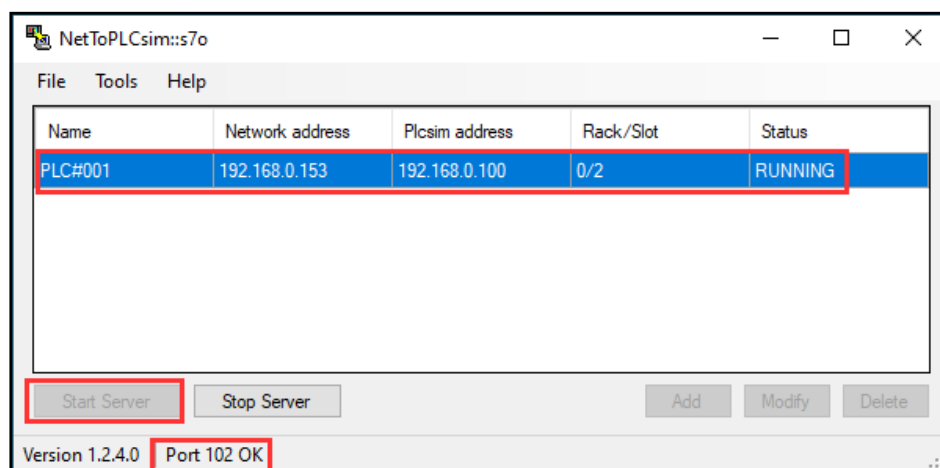
Fonte: Elaborado pelo autor

Figura 19 – NetToPLCSim



Fonte: Elaborado pelo autor

Figura 20 – NetToPLCSim

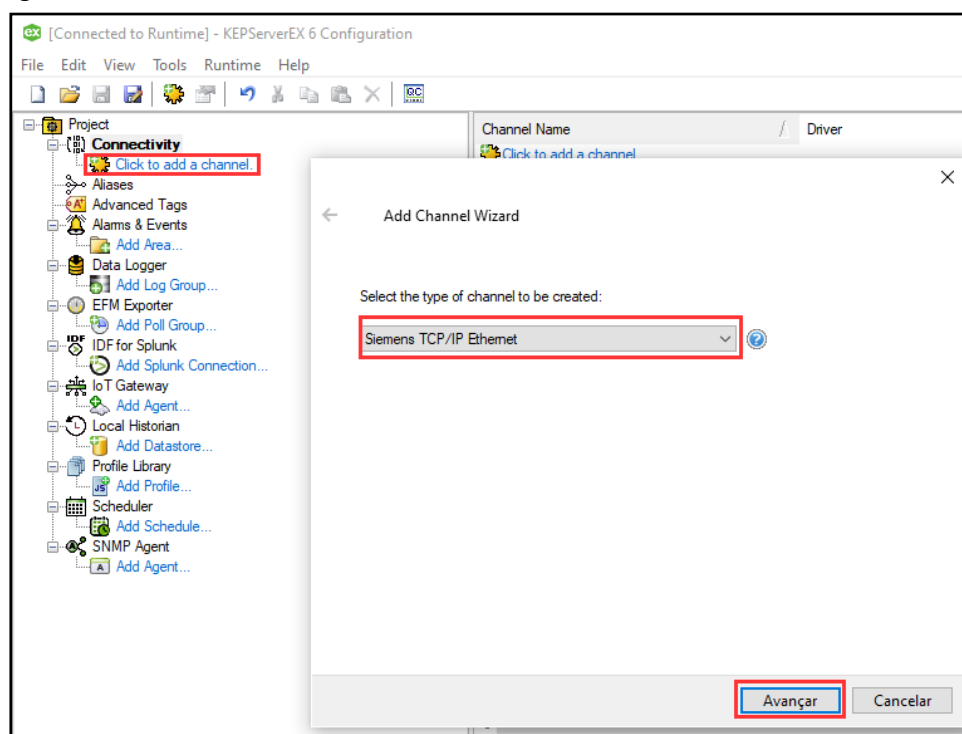


Fonte: Elaborado pelo autor

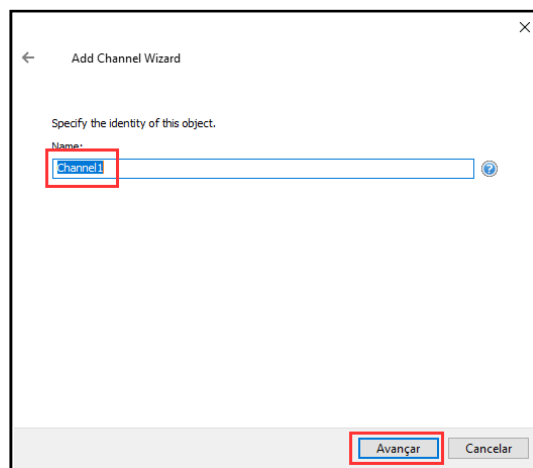
Terminado as configurações da aplicação do *NetToPLCSIM*, é necessário criar o elo de comunicação entre o PLC e o controlador do robô através do *software* KEPServerEX com o protocolo OPC.

A partir da figura 21 são mostradas as configurações dos canais, dos dispositivos e das TAG's de comunicação.

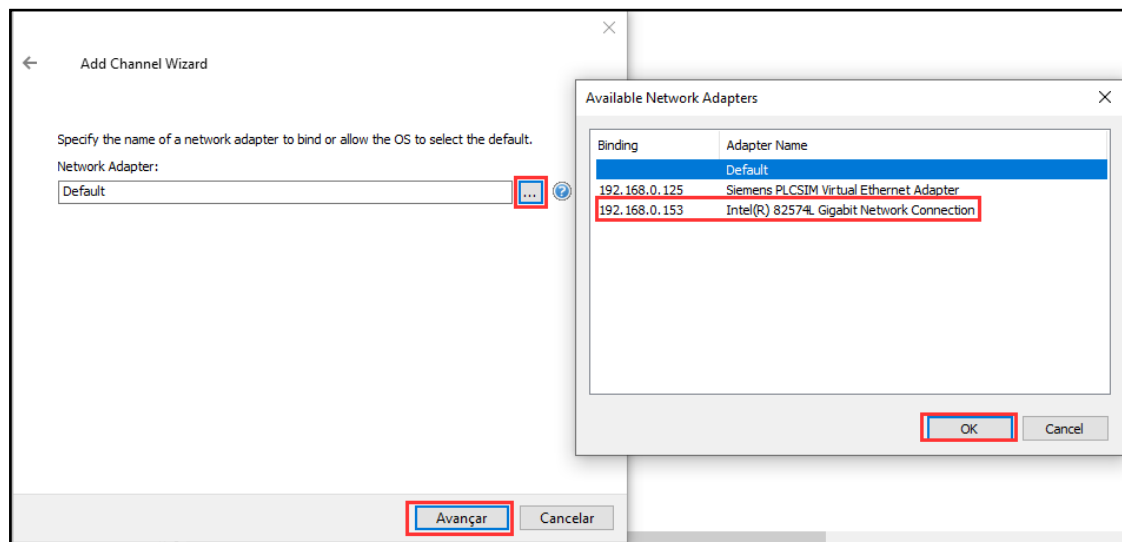
Figura 21 – KEPServerEX - Channel



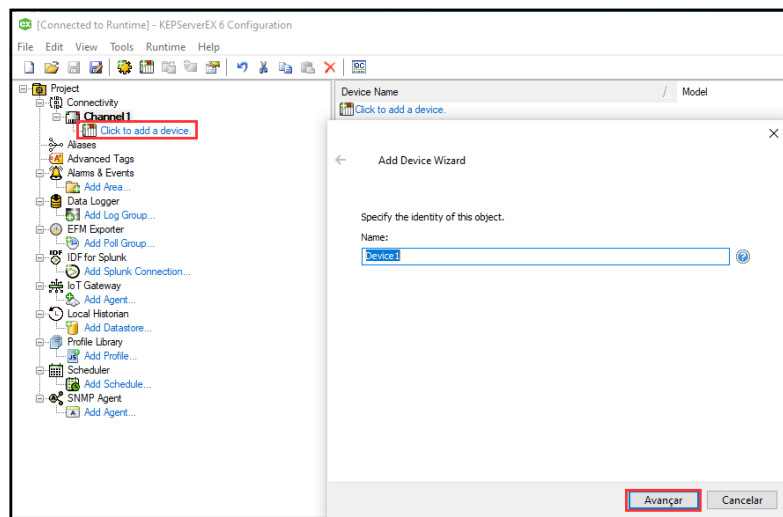
Fonte: Elaborado pelo autor

Figura 21 – KEPServerEX - Channel

Fonte: Elaborado pelo autor

Figura 22 – KEPServerEX - Channel

Fonte: Elaborado pelo autor

Figura 23 – KEPServerEX - Device

Fonte: Elaborado pelo autor

Figura 24 – KEPServerEX - Device

The figure consists of two side-by-side screenshots of the 'Add Device Wizard' dialog box. The left screenshot shows the 'Model' dropdown menu with 'S7-300' selected. The right screenshot shows the 'ID' text box with '192.168.0.153' entered. Both screenshots have red boxes highlighting the respective fields and 'Avançar' buttons.

Fonte: Elaborado pelo autor

Figura 25 – KEPServerEX - TAG

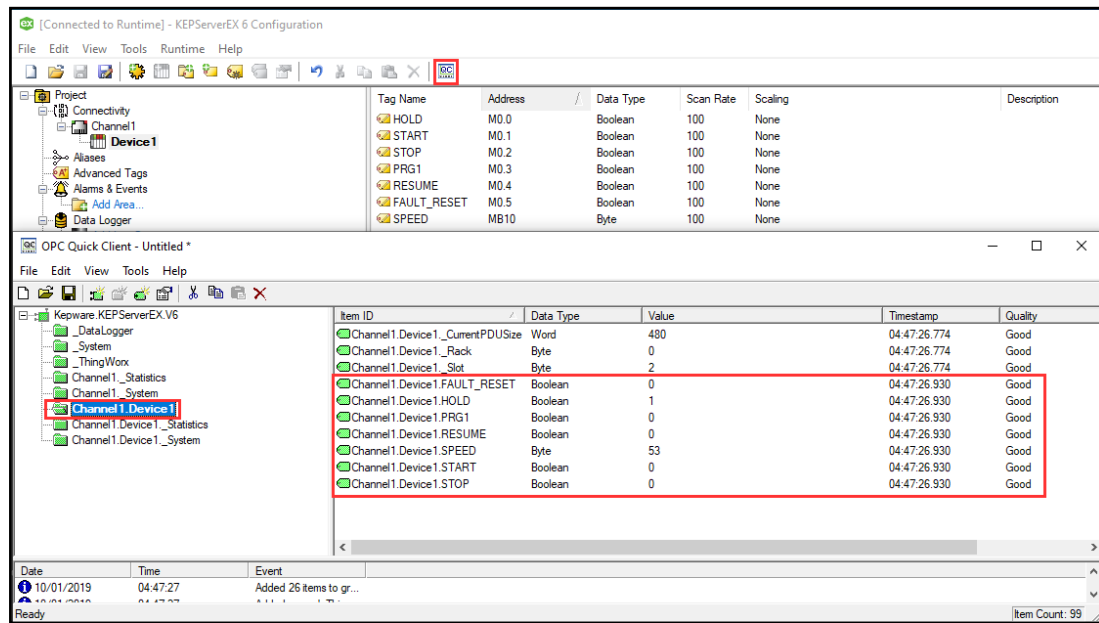
The figure shows the KEPServerEX 6 Configuration window. The left pane shows a tree view with 'Device1' highlighted. The right pane shows a table of tags.

Tag Name	Address	Data Type	Scan Rate	Scaling
HOLD	M0.0	Boolean	100	None
START	M0.1	Boolean	100	None
STOP	M0.2	Boolean	100	None
PRG1	M0.3	Boolean	100	None
RESUME	M0.4	Boolean	100	None
FAULT_RESET	M0.5	Boolean	100	None
SPEED	MB10	Byte	100	None

Fonte: Elaborado pelo autor

Ao término das criações das TAG's de interface no ambiente de configuração do *KEPServerEX*, é possível verificar o monitoramento de cada sinal em tempo real, conforme figura 26.

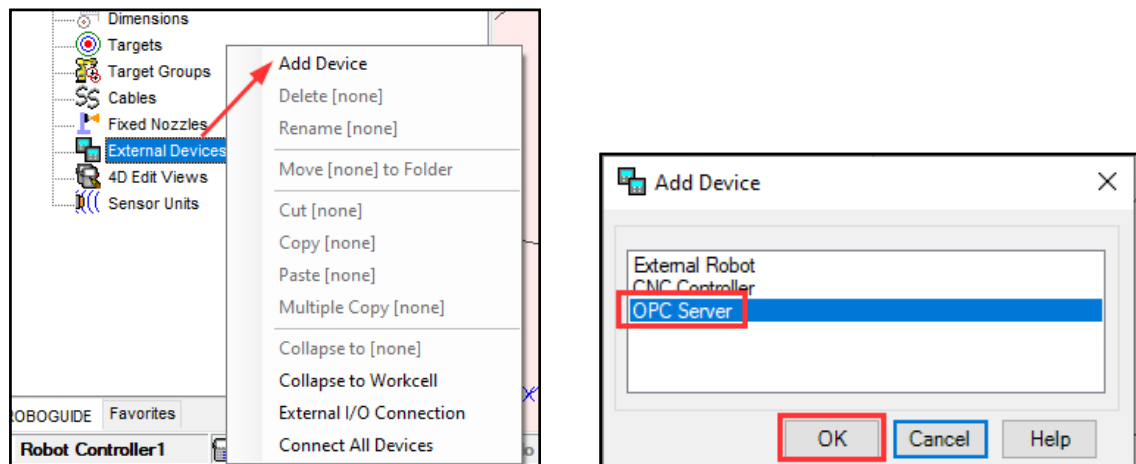
Figura 26 – KEPServerEX – monitor de sinais



Fonte: Elaborado pelo autor

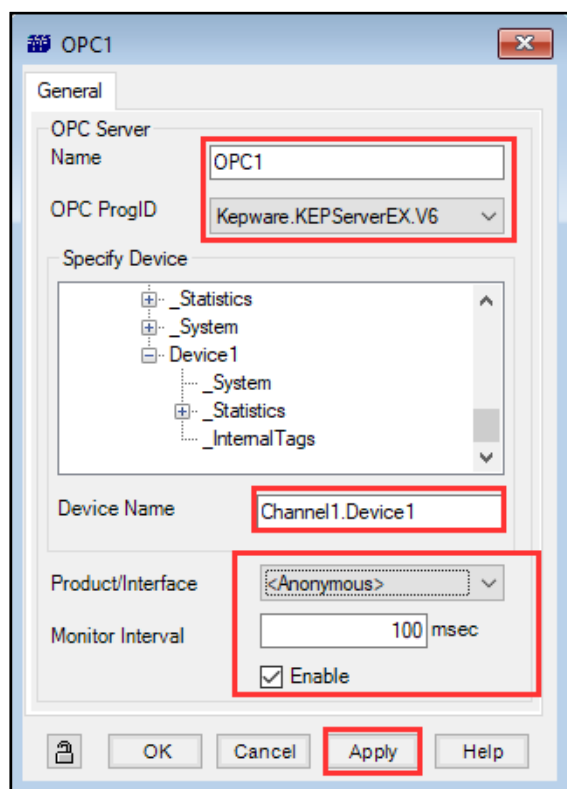
Com as TAG's de interface já criadas e configuradas, é necessário configurar a comunicação via OPC no *software Roboguide*, conforme ilustrado nas figuras 27, 28 e 29.

Figura 27 – Roboguide – Configuração do OPC



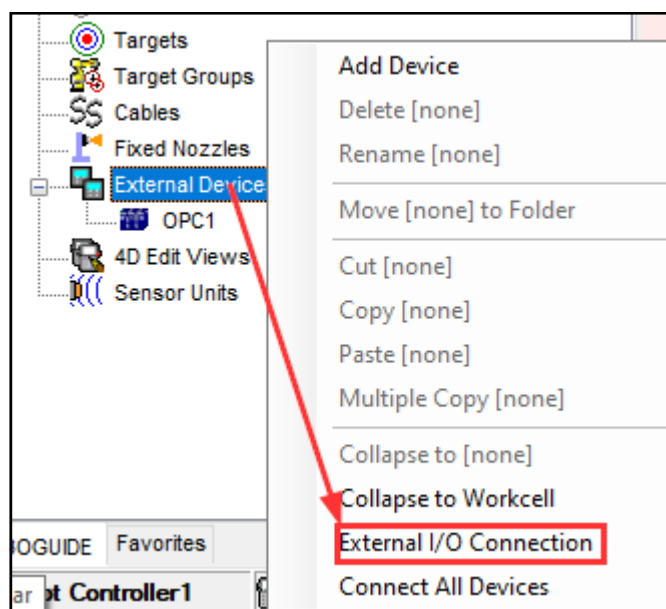
Fonte: Elaborado pelo autor

Figura 28 – Roboguide – Configuração do OPC



Fonte: Elaborado pelo autor

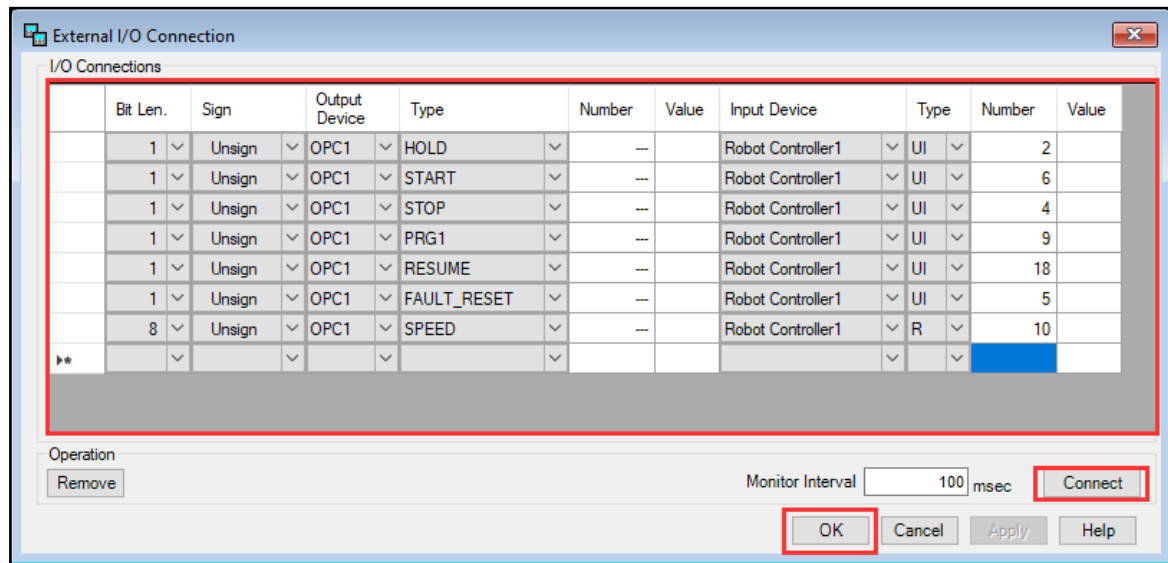
Figura 29 – Roboguide – Configuração do OPC



Fonte: Elaborado pelo autor

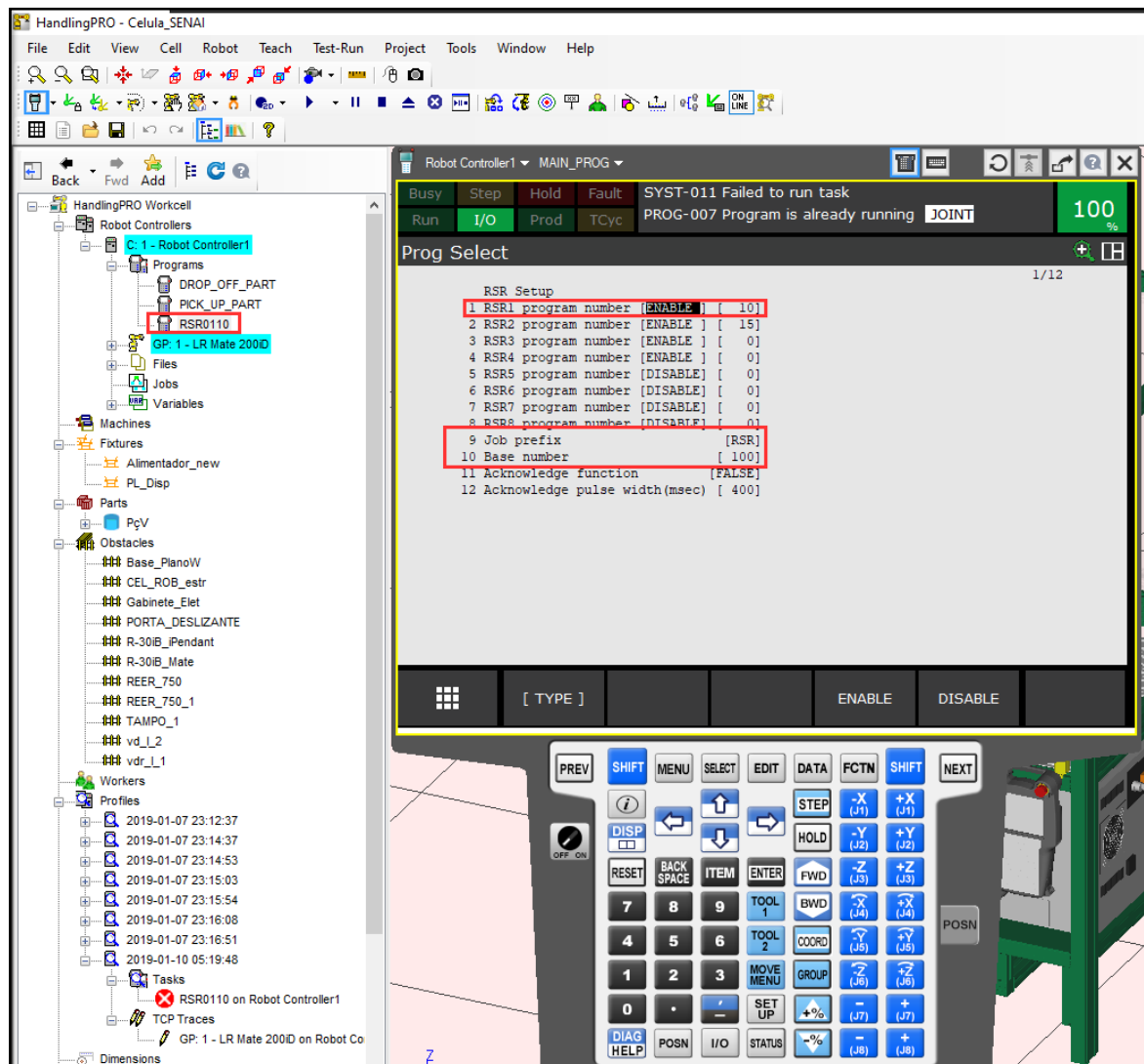
Na figura 30 está a configuração do elo de sinais entre o OPC e as entradas digitais do controlador do robô (UI). Na figura 31 está a configuração de nomeação definida do programa principal de movimentação do robô.

Figura 30 – Roboguide – Configuração do OPC



Fonte: Elaborado pelo autor

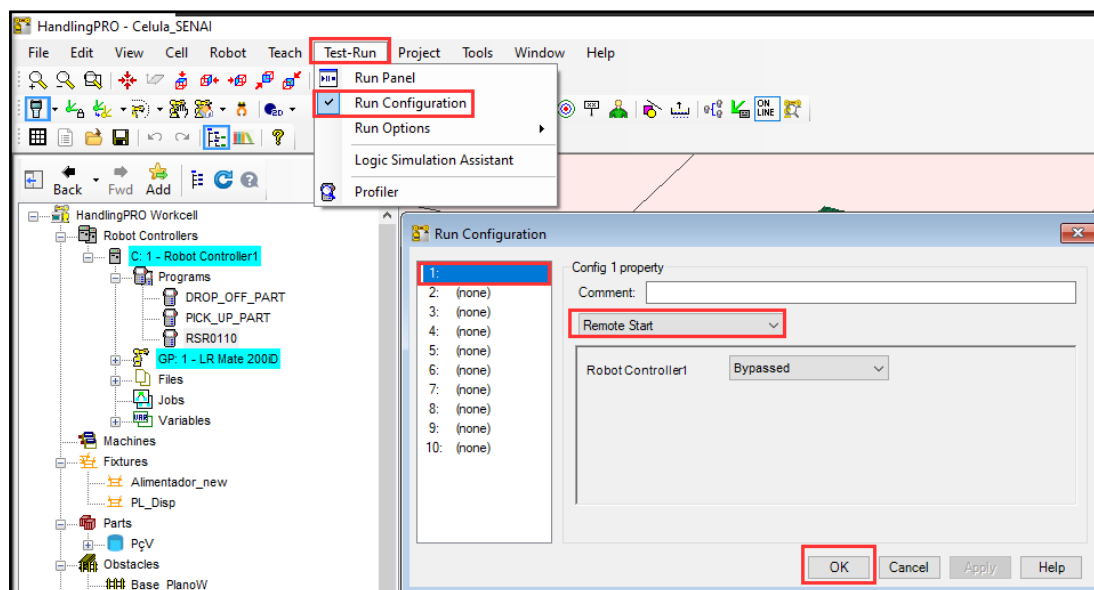
Figura 31 – Roboguide – Configuração seleção do programa principal



Fonte: Elaborado pelo autor

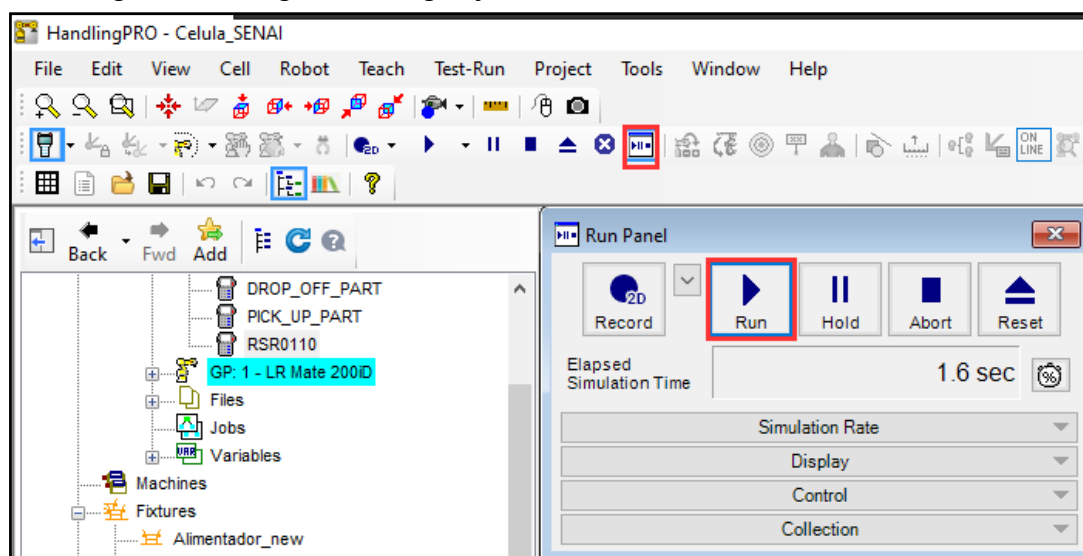
Nas figura 32 e 33 é demonstrado as configurações do controle remoto do processo de simulação do *Roboguide*.

Figura 32 – Roboguide – Configuração do controle remoto



Fonte: Elaborado pelo autor

Figura 33 – Roboguide – Configuração do controle remoto



Fonte: Elaborado pelo autor

Ao término de toda a configuração de sinais de interface para controle do robô, será possível simular a tela IHM através do *software TIA PORTAL*. Na figura 34 é mostrado como iniciar o *Runtime* de simulação da tela IHM. O usuário poderá acionar os botões de controle e verificar os sinais a interagir no programa de simulação do *Roboguide*.

Figura 34 – TIA PORTAL – Simulação da IHM



Fonte: Elaborado pelo autor

3 Considerações finais

O objetivo principal deste artigo foi demonstrar de forma simples e objetiva todos os passos necessários para estabelecer a comunicação, controle e interface entre o *Roboguide* e o *TIA PORTAL*, a possibilitar um ganho significativo na engenharia de simulação e de controle da automação.

REFERÊNCIAS

KEPServerEX. Disponível em:

<https://www.kepserverex.pt/o-que-e-o-kepserverex-como-funciona-e-como-pode-ajuda-lo/>.

Acesso em: 10 jun. 2021.

NETTOPLCSIM. Disponível em:

<http://nettoplcsim.sourceforge.net/>.. Acesso em: 10 jun. 2021.

#OPC #KEPServerEx 6 #PLCSim #ROBOGUIDE #Part #14. 1 vídeo (28 min). Publicado pelo canal PLC And ROBOT Academy. Disponível em:

<https://www.youtube.com/watch?v=Q0YvKNTInfM&list=PLU4oa1gKL-fFfPHj2vKjL4eJUCt5eDV&index=3>. Acesso em: 10 jun. 2021.

WIKIPEDIA. Boolean. 2020. Disponível em: <https://pt.wikipedia.org/wiki/Boolean>. Acesso em: 10 jun. 2021.

WIKIPEDIA. PROFINET. 2020. Disponível em: <https://pt.wikipedia.org/wiki/PROFINET>. Acesso em: 10 jun. 2021.

AGRADECIMENTOS

Primeiramente a Deus, por ter me concedido saúde e sabedoria para realizar esse projeto em um momento delicado a qual a nossa família vivenciou durante o período de pandemia.

Ao corpo docente da Faculdade SENAI de Tecnologia Mecatrônica, em especial ao professor Júlio Cesar de Almeida Freitas pela paciência, dedicação e pelo encorajamento na elaboração desse projeto. Contribuindo para o meu crescimento pessoal e profissional na área de simulação e comissionamento virtual.

A minha esposa Raquel Gregorio dos Santos e aos nossos filhos Melissa e Miguel, pela compreensão, apoio e paciência.

Dedico este artigo ao meu pai Paulo Gregorio da Silva, que me ensinou desde pequeno o caminho que deveria seguir e ao meu irmão Wilson Gregorio da Silva, por ter sido uma referência do mundo da automação para mim.

Sobre os autores:

ⁱ WILLIAM GREGORIO DA SILVA



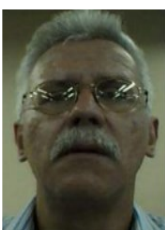
Cursando atualmente Superior em Tecnologia Mecatrônica Industrial pela Faculdade SENAI de Tecnologia Mecatrônica (2021). Formado em Técnico em Eletroeletrônica (2006) e Eletricista de Manutenção Eletroeletrônica pela Escola SENAI “A. Jacob Lafer (2001). Tem experiência na área de Automação Industrial. É líder de Automação Industrial na empresa B. GROB do Brasil responsável pelo setor de Colocação em Funcionamento.

ⁱⁱ JULIO CESAR DE ALMEIDA FREITAS



Atualmente é professor de educação profissional tecnológica, aulas na pós-graduação de indústria 4.0, na graduação em Tecnologia em Mecatrônica Industrial no SENAI Armando de Arruda Pereira. Possui graduação em Engenharia Mecânica pelo Centro Universitário da FEI (1991) e mestrado em Engenharia Mecânica pela Universidade Estadual de Campinas (UNICAMP 2004). Engenheiro consultor – Robtech. Tem experiência na área de Engenharia Mecânica, com ênfase em Robotização, atuando principalmente nos seguintes temas: integração de robô, programação on-line e off-line, soldagem robotizada. <http://lattes.cnpq.br/5398642463543389>

ⁱⁱⁱ PAULO SEBASTIÃO LADIVEZ



Possui graduação em Engenharia Elétrica pela Universidade Mogi das Cruzes (1984) com especialização em Tecnologias e Sistemas de Informação pela Universidade Federal do ABC (2013). Atualmente é professor da Faculdade SENAI de Tecnologia Mecatrônica, lecionando as disciplinas Projetos, Microcontroladores, Linguagem de Programação no Curso de Tecnologia em Mecatrônica Industrial e na Pós-Graduação em Automação Industrial. Tem experiência na área de Engenharia Eletrônica, com ênfase em Automação Industrial e Mecatrônica, atuando principalmente nos seguintes temas: Mecatrônica, Manufatura Digital, Redes Industriais, Automação Industrial, Microcontroladores e Controle.