



**Universidade Federal de Itajubá - UNIFEI**

Instituto de Engenharia de Sistemas e Tecnologia da Informação - IESTI

# **Guia de Configuração**

## Bancada Didática B&R

**Rafael Coelho Paes - 2019000081**

Orientador: Prof. Dr. Jeremias Machado Barbosa

## 1 Introdução

Este guia tem como objetivo auxiliar os alunos de automação na utilização do módulo didático de Automação Integrada B&R. Para desenvolver atividades com este sistema será necessário um computador com o software *Automation Studio* instalado. No roteiro foi utilizada a versão 4.12.2.93 para executar simulações, porém, para a aplicação na bancada física a versão 3.0.90.18 pode ser a mais adequada. Os computadores do laboratório de controle *LCPIC* já contam com a licença do software para esta versão. Caso o aluno deseje executar simulações em sua própria máquina, é possível adquirir uma versão gratuita de testes do programa válida por 90 dias no site da empresa <<https://www.br-automation.com/en/service/software-registration/automation-studio-licensing/>>

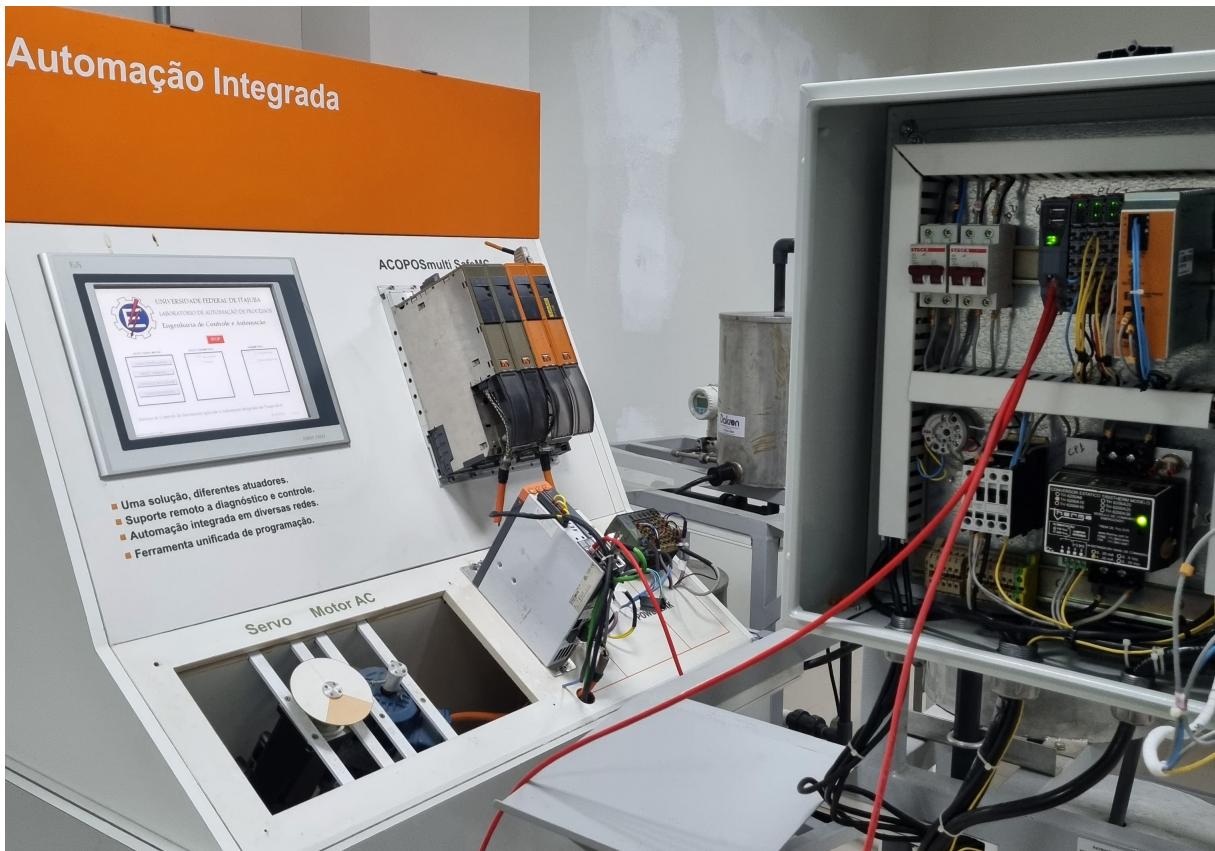


Figura 1 – Bancada B&R

## 2 Componentes Físicos

Serão necessários, para a confecção do ambiente de testes e aplicação, uma série de equipamentos. A bancada de testes pode ser visualizada na Figura (1). Os principais equipamentos de *Hardware* que serão utilizados neste guia são:

- CLP X20CP0483 (Figura 2)
- Servo Motor 8LS35E2030.D000-0 (Figura 3)
- Servo Drive ACOPos 8V1016.00-2 (Figura 4)

O drive utilizado oferece diversas opções de personalização através de módulos plug-in. Na bancada didática do laboratório, o servo drive vem equipado com os três módulos:

- **8AC114.60-2** - Interface POWERLINK V2 - x2 Ports RJ-45;
- **8AC120.60-1** EnDat 2.1 Encoder: Combinação de um Encoder absoluto e incremental;

- **8AC130.60-1** Módulo de 8 entradas/saídas digitais.

O drive manterá a comunicação com o CLP por meio do seu módulo *8AC114.60-2*, utilizando a rede **Ethernet Powerlink**, um protocolo aberto para Ethernet padrão introduzido pela B&R em 2001.



Figura 2 – CLP X20CP0483



Figura 3 – Servo motor 8LS35E2030



Figura 4 – Servo drive ACOPOS 1016

### 3 Configuração Automation Studio

#### 3.1 Setup Inicial

##### 3.1.1 Criação do Projeto

Para a criação de num novo projeto, basta clicar em **File > New Project** e uma janela como na Figura 5 irá abrir. Nomeie-o com sua preferência e vá para a próxima janela.

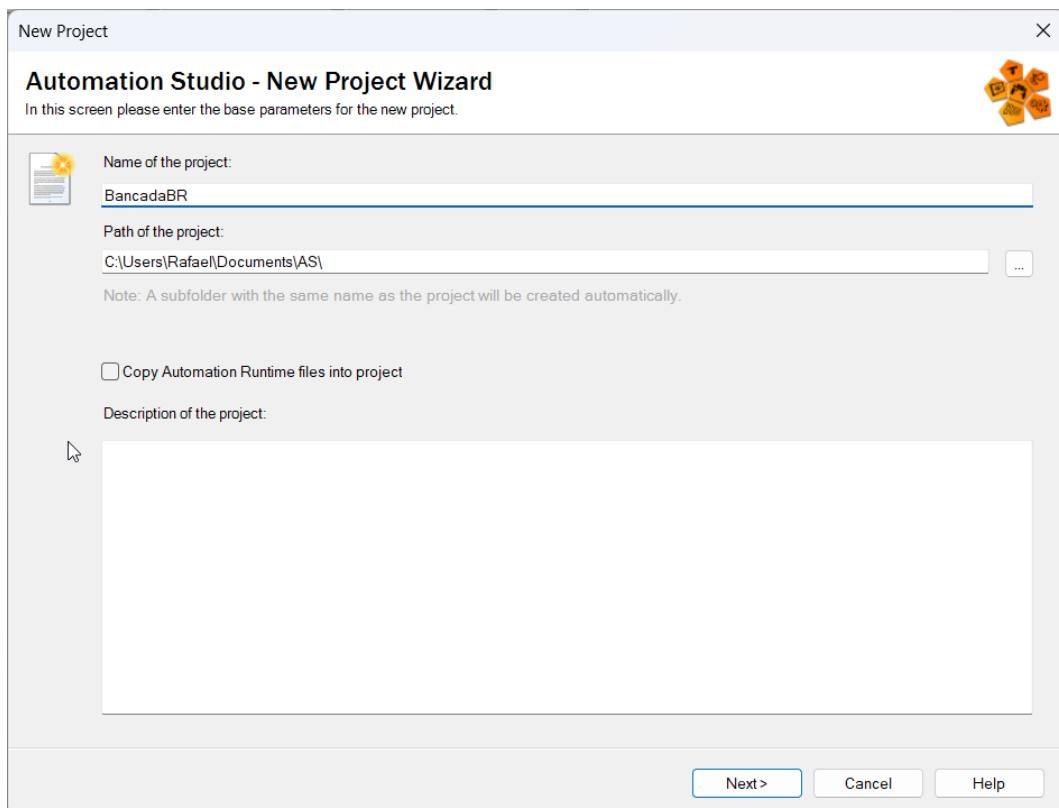


Figura 5

Na próxima página escolha a opção "Define a New Hardware Configuration Manually" e clique em *Next* (Figura 6).

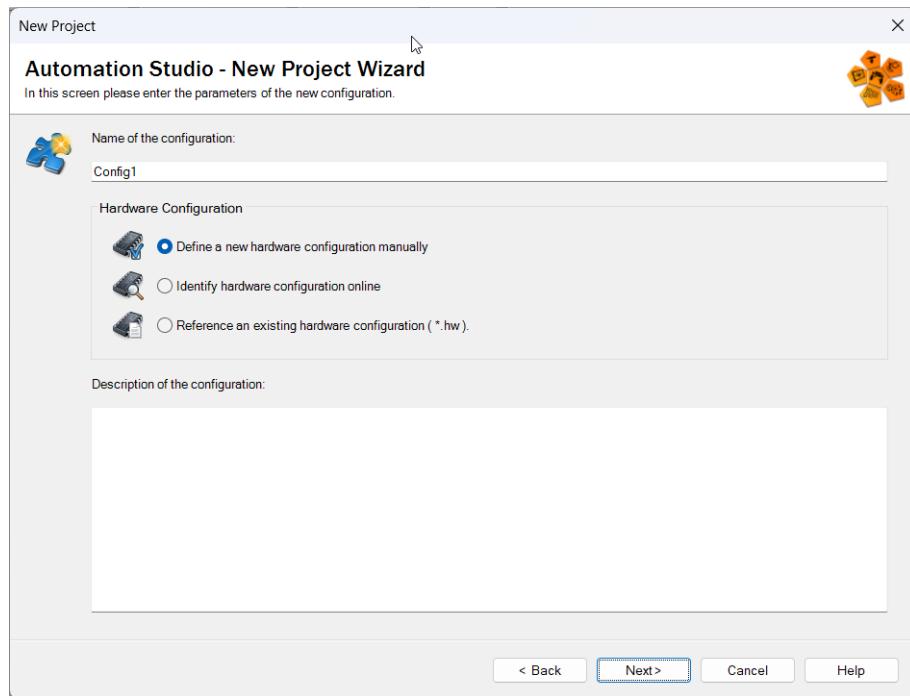


Figura 6

Escolha o CLP instalado na bancada: X20CP0483. É possível filtrar a busca digitando o nome dos equipamentos na barra superior, como na Figura 7.

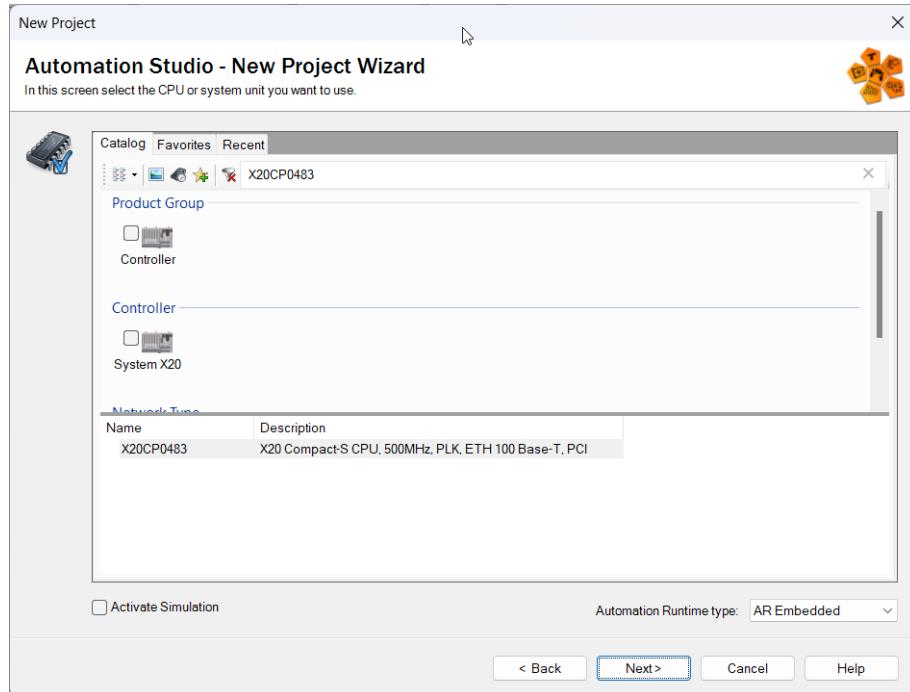


Figura 7

A próxima página pedirá que você escolha um módulo de hardware, no entanto a demonstração esta opção não será necessária. Conclua a criação do projeto clicando em *Finish*.

### 3.1.2 Incluindo Hardwares

Como explicado anteriormente, para a utilização da bancada será necessária um drive para controle do motor. Como se trata de um equipamento da família de produtos B&R, podemos facilmente encontrá-lo na aba *Toolbox*, como na Figura 8.

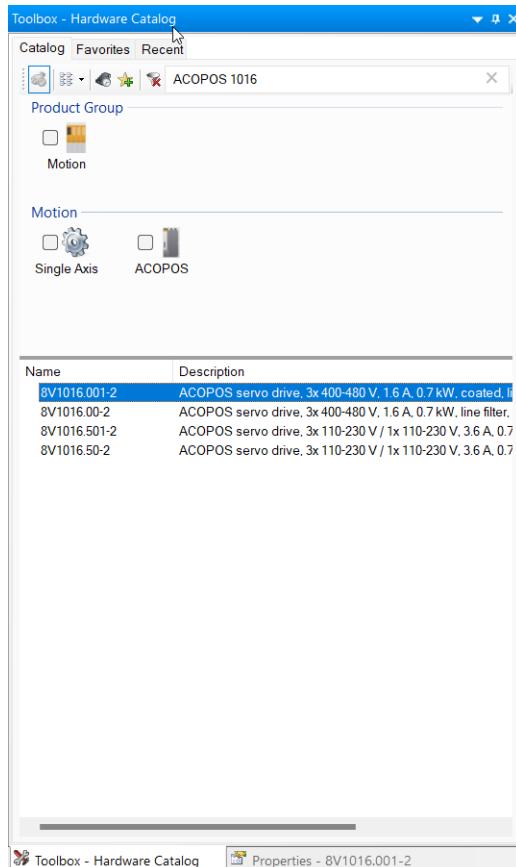


Figura 8

Além disso, será necessário incluir os módulos instalados no equipamento, também listados anteriormente. Na aba *Physical View*, selecione os slots vazios do drive e preencha-os com os módulos com auxílio da *Toolbox*. O resultado final deve se assemelhar a Figura 9.

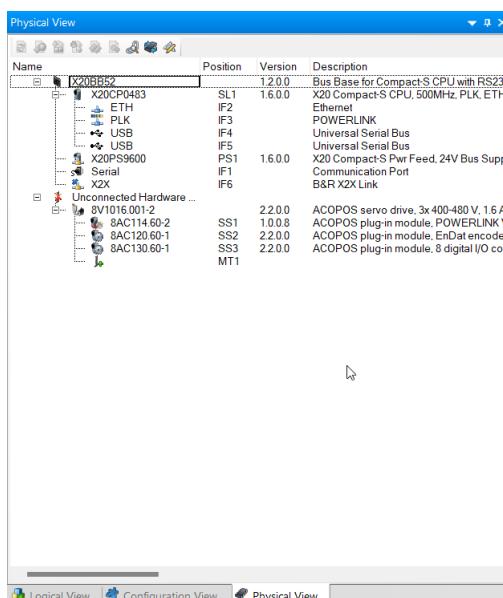


Figura 9

Agora, na janela de *System Design*, onde exibe seus Hardwares selecionados, conecte o CLP com o Drive utilizando a porta **POWERLINK** do controlador e a porta de comunicação 1 do **Módulo 8AC114.60-2**, como mostrado na Figura 10.

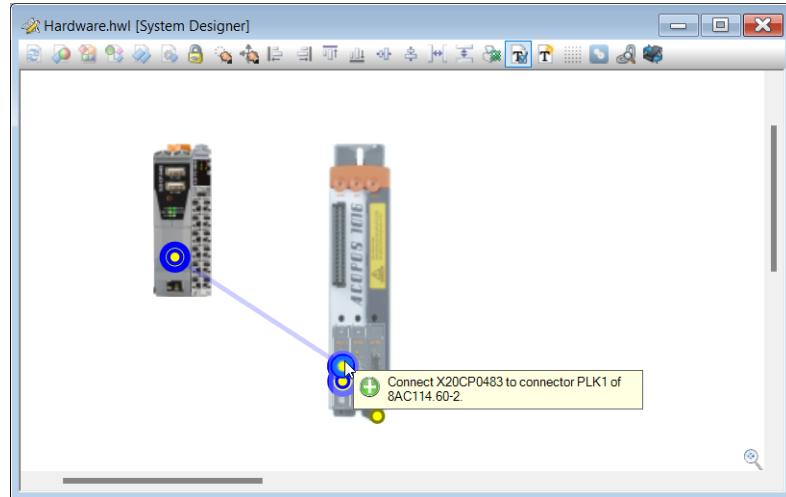


Figura 10

Ao ligar os dois componentes, uma janela se abrirá automaticamente solicitando a escolha de um motor para a aplicação de *Motion*. O motor usado na bancada não está disponível em algumas versões do Automation Studio, então será necessário escolher um que possua especificações semelhantes ao do utilizado. O modelo escolhido foi o **8LSA35.EA030D000-0** Figura 11. Quando o motor for selecionado, uma nova página aparecerá perguntando o sistema de movimentação que você deseja utilizar. Escolha a opção **mappMotion** (Figura 12). A janela de *System Designer* deve se assemelhar à da Figura 13.

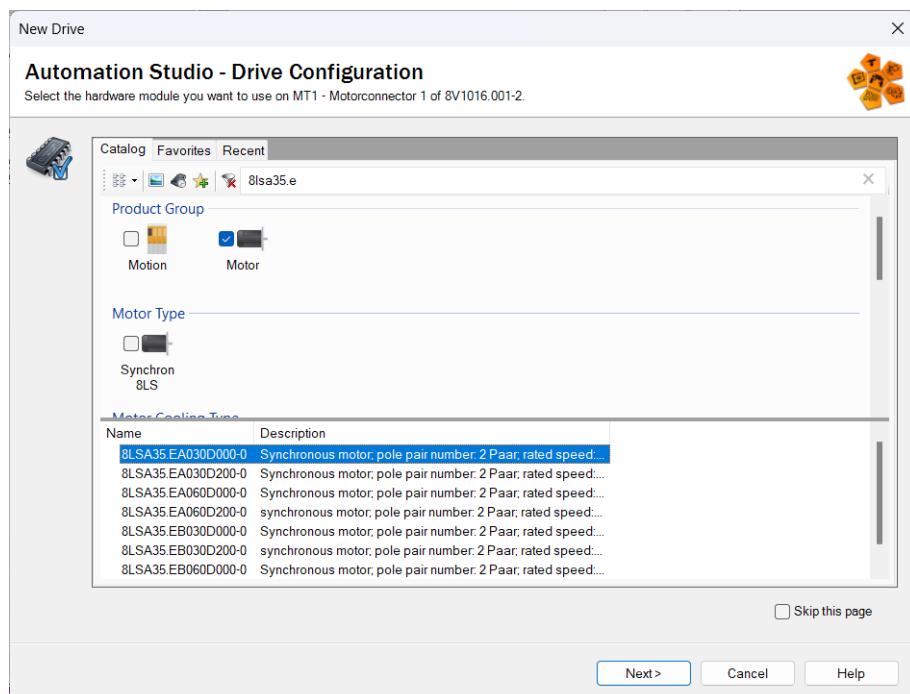


Figura 11

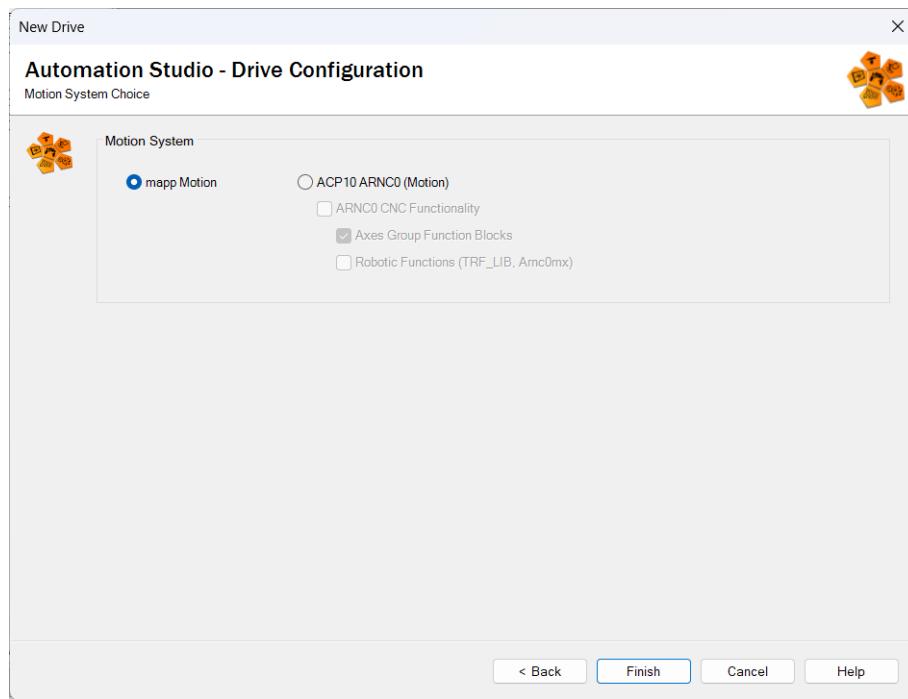


Figura 12

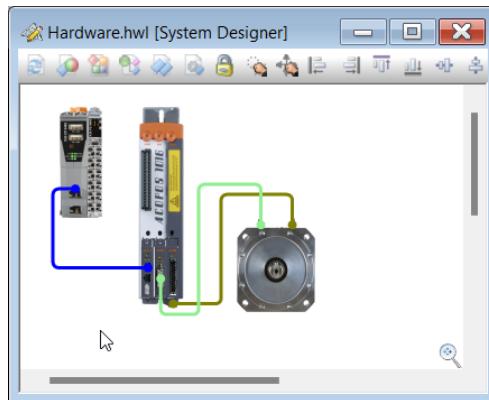


Figura 13

Ao conectar todos os módulos e equipamentos corretamente, a aba de *Physical View* estará como na Figura 14, onde o driver estará conectado ao barramento **PLK** do CLP, como desejado.

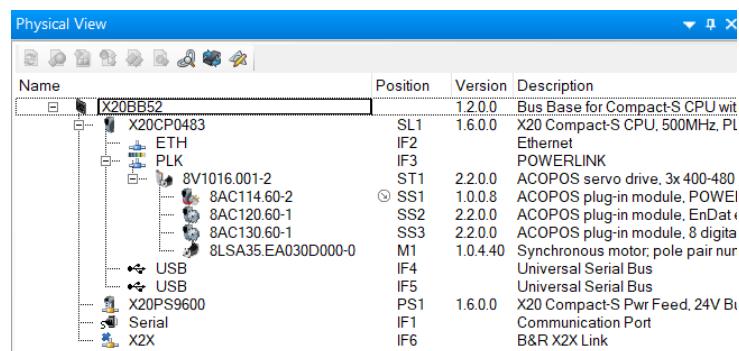


Figura 14

Uma configuração importante que deve ser feita é a seleção do *Node* do drive. O valor do *Node* configurado no *Automation Studio* deve estar correspondente ao valor do *Node* na chave física do drive para que a comunicação seja realizada.

O *Node* é uma variável de um *byte - 8 bits*, podendo, na teoria, assumir 256 valores. O posicionamento das chaves nos possibilita selecionar o valor em **Hexadecimal**, ou seja, no formato **0xYZ**. A primeira chave dita o primeiro dígito (Y, 4 bits mais significativos), enquanto a segunda dita o segundo (Z, 4 bits menos significativos). **Atente-se** ao *node* selecionado, pois alguns valores são reservados pela fabricante, como citado no **manual do módulo**:

In principle, node numbers between **0x01** and **0xFD** are permitted. However, node numbers between **0xF0** and **0xFD** are intended for future system expansions. To ensure compatibility, these node numbers should be avoided. Node numbers **0x00**, **0xFE** and **0xFF** are reserved and may therefore not be set.

Uma foto tirada do módulo utilizado na bancada é exibida na Figura 15, onde o número do módulo selecionado em hexadecimal é **0x02**, que equivale a **2** em decimal. Portanto é necessário escrever este valor nas configurações do programa. Para fazer isso, basta selecionar o módulo de comunicação **POWERLINK** com o botão direito e clicar em **Node Number > Change Node Number**, como na Figura 16.



Figura 15

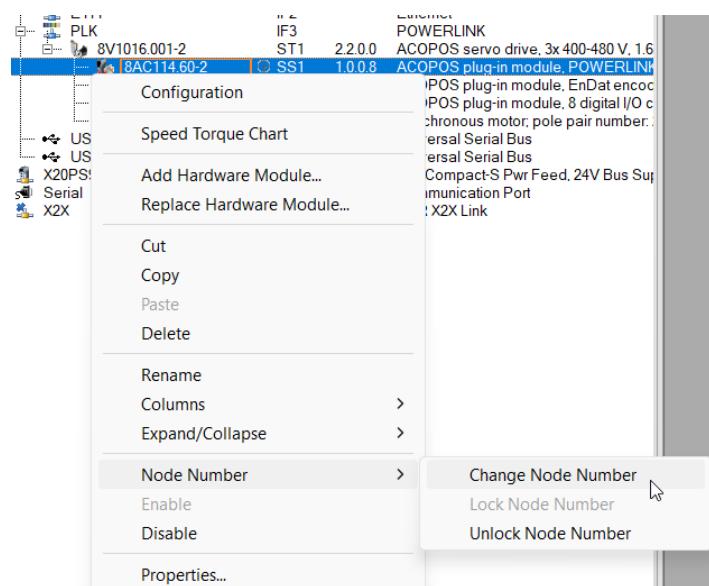


Figura 16

### 3.2 Configurando os dispositivos

Primeiramente iremos configurar o CLP clicando com o botão direito no controlador pela *Physical View* e selecionando "Configuration". No menu, a opção mais importante no momento é ativar a comunicação OPC UA, que se localiza na última linha da coluna (*OPC-UA System*). Caso desejar fazer uma simulação ao invés de uma aplicação real, basta acionar a simulação na linha *Simulation*, aba *Activate simulation*. Conferir a Figura 17 para se guiar.

Name	Value
X20CP0483	
Configuration ID	BancadaBR_Config1
Configuration version	1.0.0
Module system on target	
Simulation	
Activate simulation	off
IP address	127.0.0.1
Port number	11160
INA node number	1
Simulate OPC UA devices	on
Memory configuration	
System	
Reboot	
Communication	
Timing	
Resources	
File devices	
Time synchronization	
Internet file system	
Ethernet parameters	
DNS parameters	
Online parameters	
FTP Server	
TFTP Server	
Additionally supported hardware	
Web Server	
System diagnostics	
DTM Server	
Activate DTM Diagnostic Server	off
OPC-UA System	
Activate OPC-UA System	on

Figura 17

Partindo para a configuração do drive, abra a janela de configuração de maneira semelhante à etapa anterior. Selecione, em *Power Supply*, a opção **ETA system (For training only)**, para indicar que o drive está sendo alimentado por uma fonte 24V<sub>DC</sub> (Figura 18).

Name	Value
8V1016.001-2	
Drive configuration	
Module	
Activate ACOPOS simulation on PLC	Off
Power supply	ETA system (for training only)
Channel 1	
Real axis	
Axis reference	
Mechanical elements	
Encoder link	
Motor and position encoder	One encoder Encoder SS2.X11
Controller	
Homing	
Stop reaction	
Movement error limits	
Jerk filter	Not used
Zero vibration filter	Not used
Digital inputs	
Simulation	
Axis features	
Feature reference 1	
Virtual axis	Not used
External encoder axis	Not used
Channel features	
Feature reference 1	

Figura 18

### 3.3 Criação dos Eixos

Para executar ações de movimento por um programa, devemos criar um *Objeto* de eixo e referenciá-lo posteriormente. Para fazer isso, clique em *Configuration View*, no menu a esquerda, e selecione a pasta de *mappMotion* (Figura 19). Note que a aba de *Toolbox* irá mudar para exibir os itens relacionados a controle de movimento. Procure e clique duas vezes no objeto *Axis*. Caso necessário, use os filtros para facilitar a busca (Figura 20).

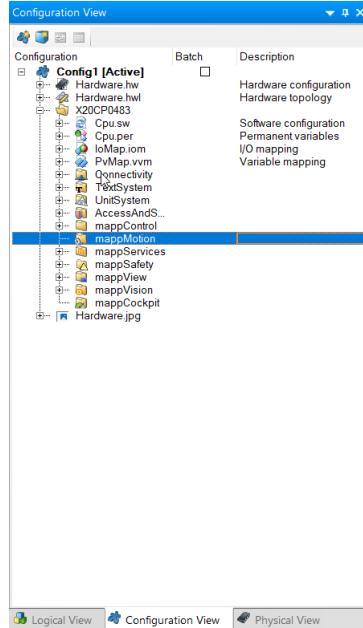


Figura 19

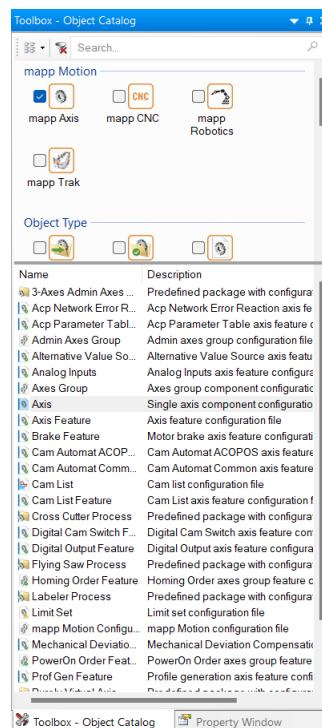


Figura 20

O eixo adicionado terá como nome padrão *gAxis\_1*, e pode ser encontrado na pasta de *mappMotion*, dentro do objeto *Config\_1.axis*. Clique duas vezes nele para abrir a janela de configuração.

Configure-o da maneira que faça mais sentido para a aplicação. Uma sugestão de configuração está disponível na Figura 21. **Atente-se aos limites de velocidade e aceleração do motor utilizado!**

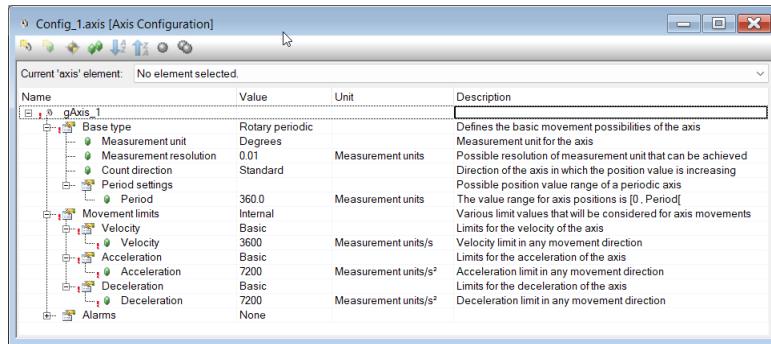


Figura 21

Ao criar o eixo no programa, é importante referenciá-lo na configuração do drive, para que o mesmo consiga efetuar o controle de movimento do motor. Para isso, basta abrir as configurações do drive na aba *Physical View* (como anteriormente) e adicionar o eixo criado na opção *Axis Reference*, como mostrado na Figura 22.

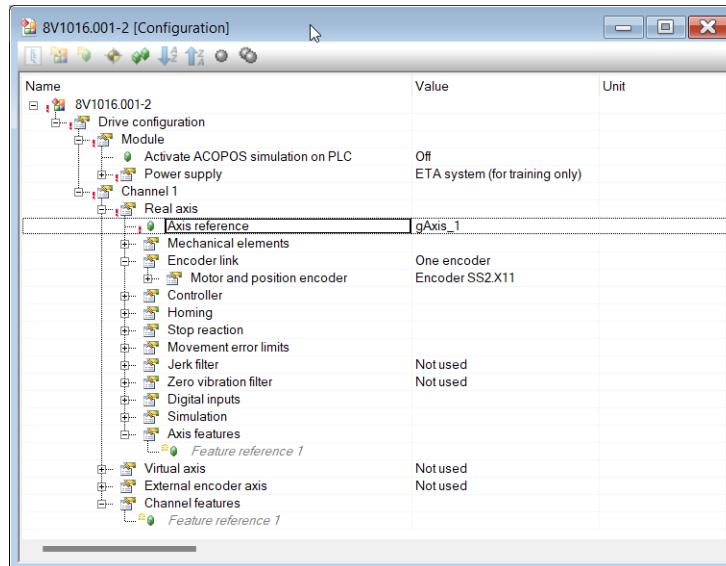


Figura 22

### 3.4 Programação

Partindo para a criação do programa, navegue para a *Logical View* e, em *Toolbox*, escolha o tipo de linguagem de sua preferência. Este tutorial mostrará uma aplicação utilizando **Texto Estruturado** (*Structured Text*, ou *ST*). É importante, no programa, criar rotinas de inicialização, loop e de finalização. O *Automation Studio* fornece estruturas de programas contendo todas estas etapas, são os chamados **All in One** (Figura 23).

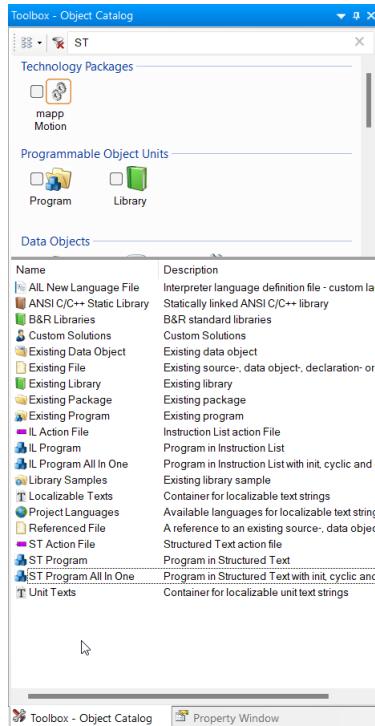


Figura 23

Ao criar um programa, clique duas vezes nele para acessá-lo. Primeiramente, será necessário adicionar duas variáveis, sendo uma delas um **Blocos Funcionais (FB)** e a outra uma **Structure**, ambos da biblioteca de *motion*.

Clique no item *Variables.var* que está dentro do programa criado e adicione duas variáveis na janela que abrir. Nomeie-as de acordo. Para selecionar o tipo, clique duas vezes no valor que está na coluna *Type* e nos três pontos e uma janela irá abrir oferecendo mais opções de tipos (Figura 24).

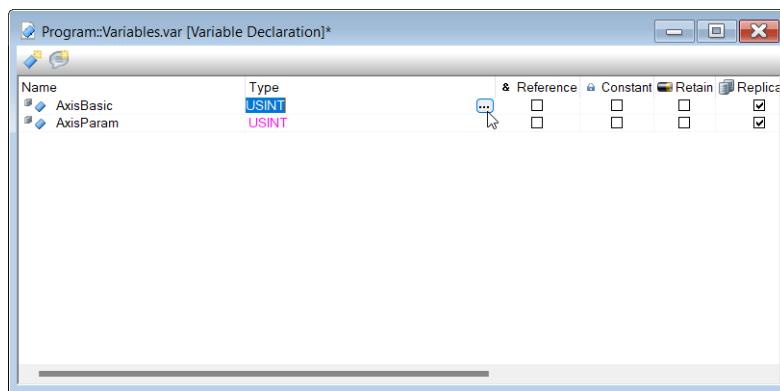


Figura 24

Escolha os tipos **MpAxisBasic** para o bloco funcional e **MpAxisBasicParType** para a estrutura (que tem como função parametrizar o motor). Use a seleção de categorias para encontrar os tipos mais facilmente, como nas Figuras 25 e 26.

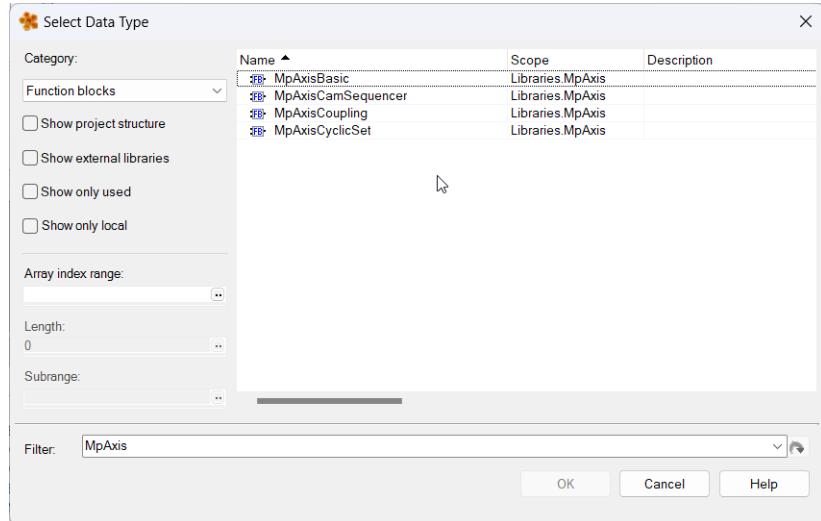


Figura 25

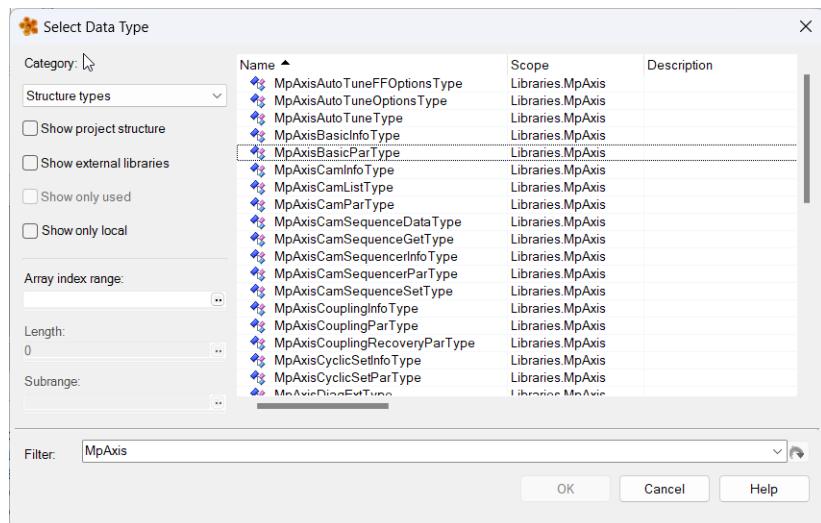


Figura 26

Por fim, sua lista de variáveis inicial deve estar como na Figura 27. Se preferir, é possível também apenas escrever o nome do tipo no campo da tabela, de forma a definir os tipos mais rapidamente que o método apresentado anteriormente.

Name	Type
AxisBasic	MpAxisBasic
AxisParam	MpAxisBasicParType

Figura 27

Partindo para o programa de uma aplicação muito básica de movimentação, basta efetuar a chamada do bloco funcional com algumas variáveis de entrada. No caso, o **endereço** do eixo criado e o **endereço** da estrutura de parâmetros recém criada devem ser as entradas de *MpLink* e de *Parameters* do bloco, respectivamente. A entrada *Enable* deve ser **TRUE** durante todo o loop do programa, e

**FALSE** quando o CLP parar de executá-lo (por isso na rotina de **EXIT** o bloco é chamado novamente com a alteração do valor). O programa descrito está sendo exibido na Figura 28.

```

st Program:Main.st [Structured Text]
PROGRAM _INIT
END_PROGRAM

PROGRAM _CYCLIC
    AxisBasic(MpLink := ADR(gAxis_1), Enable := TRUE, Parameters := ADR(AxisParam));
END_PROGRAM

PROGRAM _EXIT
    AxisBasic(MpLink := ADR(gAxis_1), Enable := FALSE, Parameters := ADR(AxisParam));
END_PROGRAM

```

Figura 28

Não se esqueça de configurar a prioridade e tempo de execução do programa criado pelo CLP. Para isso, clique duas vezes no dispositivo em *Physical View* e mova o programa para o *Cyclic #1 - [10 ms]*, como na Figura 29. **Programas de motion devem sempre estar sendo executados pelo ciclo mais rápido do controlador.**

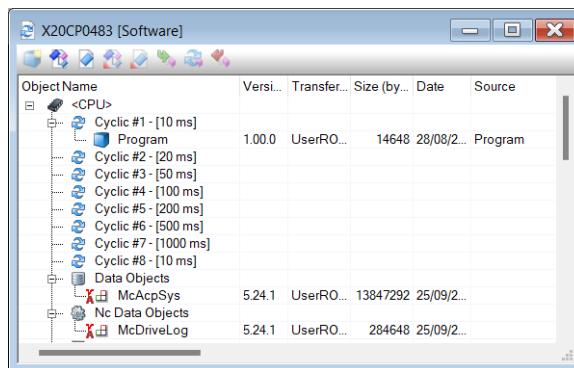


Figura 29

Parabéns! Agora será possível utilizar de maneira prática a bancada didática B&R, que oferece diversas possibilidades de aprendizado na área de controle e automação. Caso alguma configuração adicional seja necessária, consulte os próximos tópicos deste guia.

### 3.5 Testes

Para efetuar os testes (em bancada ou em simulação), basta ativar o modo de simulação do programa clicando no ícone do semáforo no painel superior (Figura 30) e depois clicar em *Build* para compilar o projeto (Figura 31).

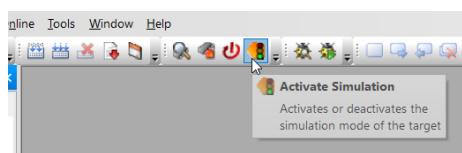


Figura 30

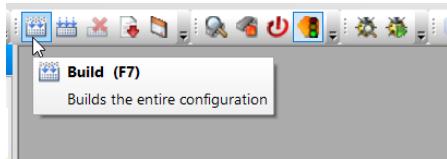


Figura 31

Ao final da compilação, se tudo correr bem, será perguntado se o usuário deseja transferir o programa para o Target, confirme, e então a tela da Figura 32 irá abrir. Clique em *Transfer* e aguarde a transferência do programa para o dispositivo (real ou simulado).

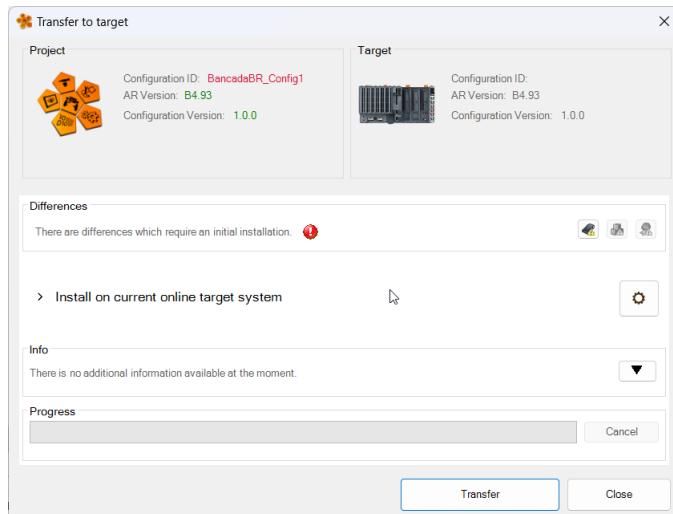


Figura 32

Com as variáveis criadas é possível testar o movimento do eixo do motor. Isso pode ser feito através da programação do *script* criado ou forçando as variáveis na tela de *Watch*, que pode ser aberta ao selecionar o objeto *Variables.var* com o botão direito e clicando em ***Open > Watch***.

Adicione as variáveis criadas para manipular o eixo alterando os valores contidos nas estruturas das variáveis. Um exemplo é acionar a variável **Power**, depois ativar e desativar a variável **Home**, e em seguida ativar **MoveVelocity** (Figura 33). Isso fará o eixo rotacionar indefinidamente com a aceleração e velocidade configurada na variável de tipo *MpAxisBasicParType*.

Name	Type	Scope	Force	Value
AxisBasic	MpAxisBasic	local		
MpLink	UDINT			106442648
Enable	BOOL			TRUE
ErrorReset	BOOL			FALSE
Parameters	UDINT			110234592
Update	BOOL			FALSE
Power	BOOL			TRUE
Home	BOOL			FALSE
MoveVelocity	BOOL			TRUE
MoveAbsolute	BOOL			FALSE
MoveAdditive	BOOL			FALSE
Stop	BOOL			FALSE
JogPositive	BOOL			FALSE
JogNegative	BOOL			FALSE
LimitLoad	BOOL			FALSE
ReleaseBrake	BOOL			FALSE
Simulate	BOOL			FALSE
AutoTune	BOOL			FALSE
Active	BOOL			TRUE
Error	BOOL			FALSE
StatusID	DINT			0
UpdateDone	BOOL			FALSE
Position	LREAL			193.57
Velocity	REAL			5.0
CommandBusy	BOOL			TRUE
CommandAborted	BOOL			FALSE
PowerOn	BOOL			TRUE
IsHomed	BOOL			FALSE
InVelocity	BOOL			TRUE
InPosition	BOOL			FALSE
MoveActive	BOOL			TRUE
MoveDone	BOOL			FALSE
Stopped	BOOL			FAT SE

Figura 33

Em caso de erro (Variável *Error* em **TRUE**), muitas vezes é possível *resetar* o estado de erro através da variável *ErrorReset*, acionando-a e desativando-a logo em seguida. Se o erro persistir, o erro presente é grave e é recomendado analisar o programa para encontrar uma solução.

Existem diversas variáveis para executar o controle de movimento do motor. **MoveAbsolute** move para a posição dada pela variável de *Position* da variável de parâmetros, e **MoveAdditive** move o eixo no valor da variável *Distance*, também da variável de parâmetros. É possível, também, executar um **Jog** e **Homing**. Se necessário, o bloco funcional também possibilita executar o **Tuning** do motor para identificação do sistema de controle.

Todos esses comandos podem ser executados por uma interface gráfica *web* desenvolvida pela B&R, chamada *mappCockpit*. Para criá-la, basta adicionar um objeto de configuração na pasta *mappCockpit*, localizado na *Configuration View*. Caso queira performar testes, altere a permissão de escrita para *Everyone* para que não seja necessário *logar* no sistema para efetuar a alteração das variáveis. Após recompilar e passar a nova configuração para o CLP, basta abrir o site <http://<PLC IP or hostname>:8084/mappCockpit/index.html> em um navegador. Também é possível gerar gráficos pela plataforma com a função *Trace*, que não será cobrida neste guia.

A Figura 34 exibe *Printscreen* da tela da interface gráfica (Aparência pode diferir dependendo da versão do *Automation Studio* instalada).

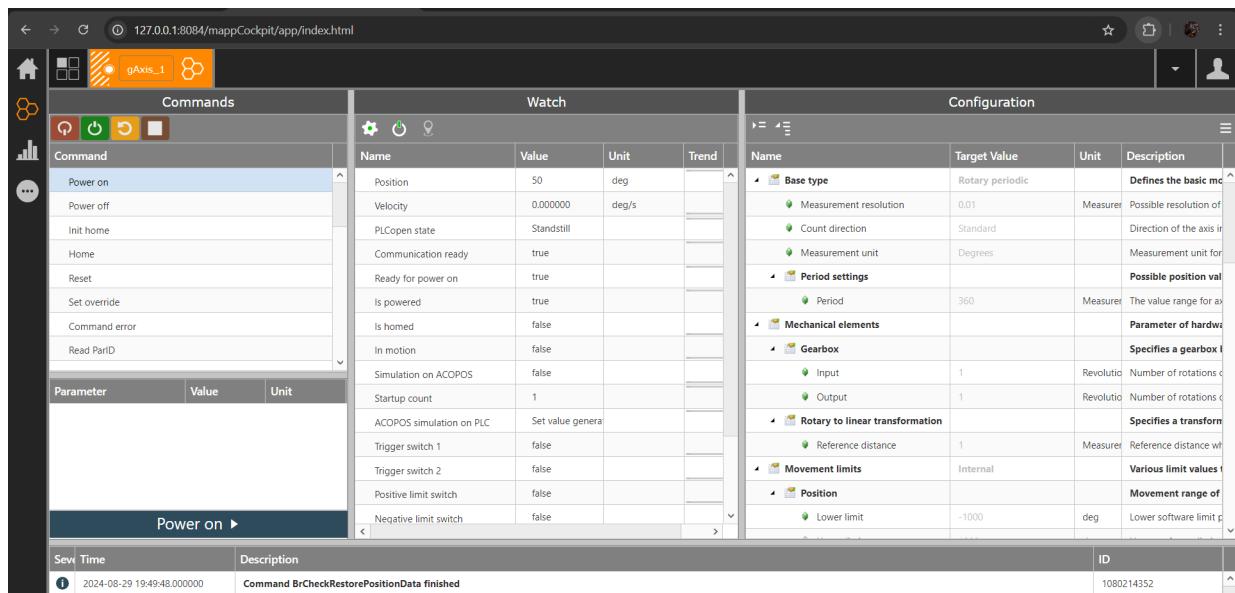


Figura 34

### 3.6 Configuração OPC UA

Se a disponibilização das variáveis em um servidor OPC UA for desejada, é possível atingir este objetivo por meio da adição de um objeto OPC UA que irá listar as variáveis disponíveis no servidor. Semelhante ao processo feito integração do *mappCockpit*, em *Configuration View*, na pasta **Connectivity > OPC UA**, crie o objeto *OPC UA Default View File*.

Neste objeto, adicione as variáveis que você deseja colocar no servidor OPC UA selecionando-as e clicando no símbolo verde *Enable Tag* (Figura 35). Atente-se que, para variáveis com estruturas mais complexas, deve-se especificar qual "sub-variável" deseja. Exemplo: na Figura 35, selecionar apenas "AxisBasic" não é suficiente para adicionar as demais variáveis, deve-se selecionar (individualmente ou em grupo usando *CTRL* ou *SHIFT*) todas as que você deseja dentro da estrutura. Não se esqueça de recompilar o código após efetuar a configuração para que as variáveis fiquem disponíveis no servidor.

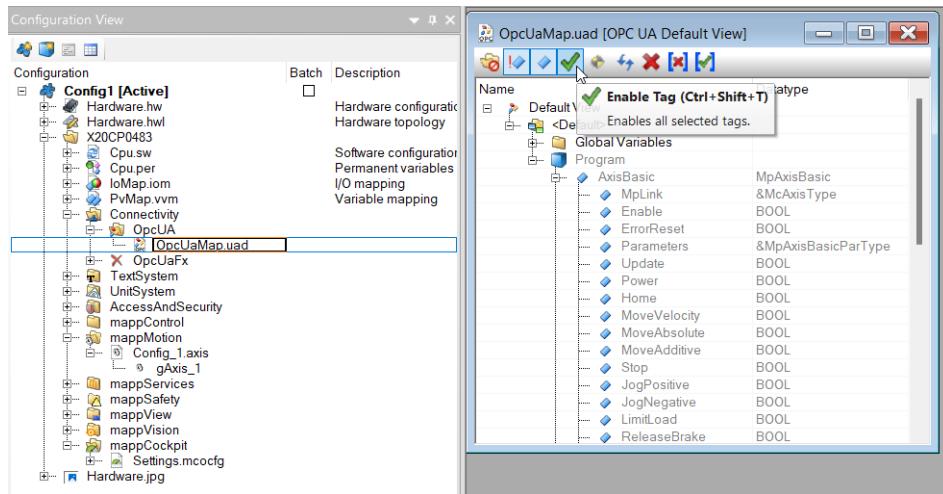


Figura 35

## 4 Conclusão

Este guia de laboratório de automação foi desenvolvido para fornecer um recurso prático e de fácil entendimento para todos os alunos que desejam aprofundar seus conhecimentos na área e para professores que desejam utilizar a bancada para fins didáticos e científicos.

Gostaria de agradecer os professores **Jeremias Machado Barbosa** e **Luiz Lenarth Gabriel Vermaas** por todo o apoio e suporte, não apenas durante o aprendizado sobre o funcionamento da bancada, mas também durante todos os semestres de minha graduação.

Caso haja alguma dúvida em relação à bancada ou sobre a programação descrita no guia, sinta-se livre para me enviar uma mensagem no email: [rafaelcpaes@unifei.edu.br](mailto:rafaelcpaes@unifei.edu.br), que tentarei responder. Para acessar o projeto base desenvolvido no guia, visite o repositório no *GitHub*, disponível no link abaixo: [<https://github.com/Rafael-CP/Guia\\_Bancada\\_BR>](https://github.com/Rafael-CP/Guia_Bancada_BR)

Continuem explorando, experimentando e inovando no campo da automação. Agradeço seu interesse e desejo muito sucesso em seus estudos e projetos futuros!