

# Procedimento e detalhes necessários para configuração, manutenção e customização do modelo MatLab/Simulink

## 1. OBTENÇÃO DO MODELO FEITO ATRAVÉS DO MATLAB

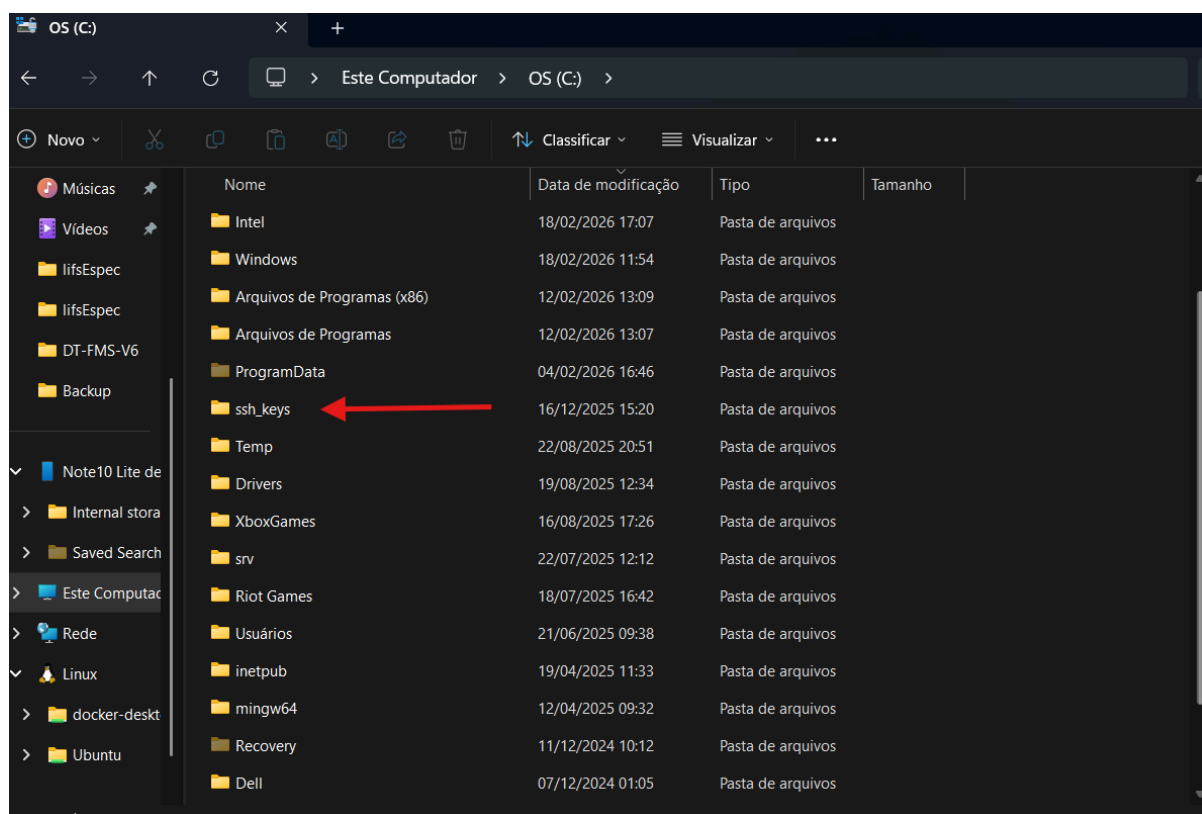
O link para obter o modelo das estações é feito através desse link a seguir:

<https://github.com/gasiepgodoy/DT-FMS/archive/refs/heads/Digital-Model---State-Transition-Table.zip>

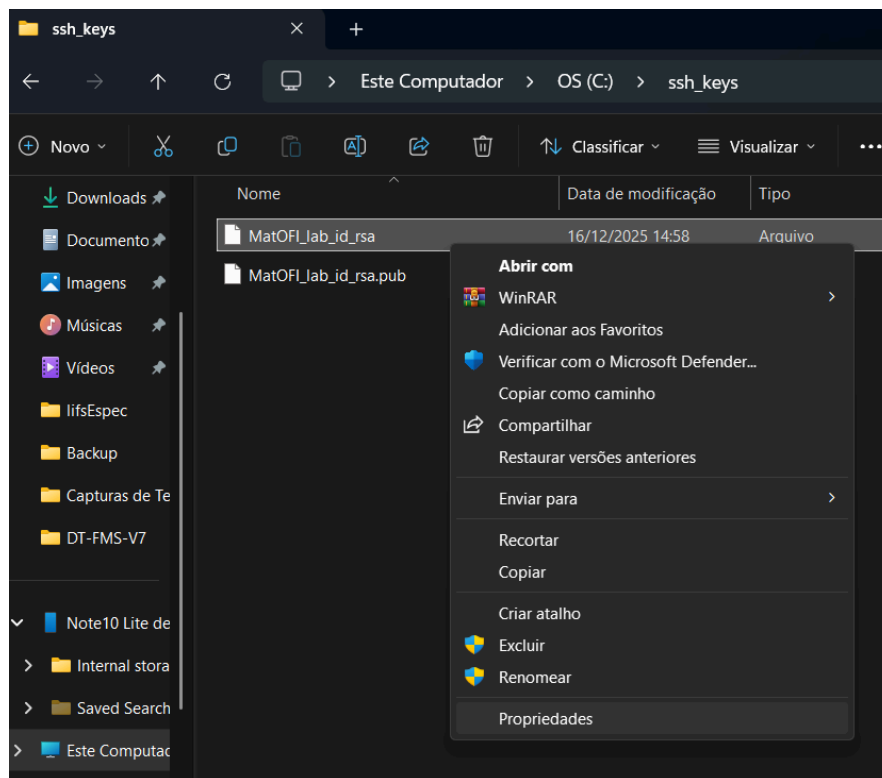
## 2. CONFIGURAÇÃO DO SISTEMA OPERACIONAL

Antes de executar qualquer script necessário para o modelo no matlab/simulink ou até mesmo o próprio modelo. É necessário a configuração de conexão da máquina que rodará o modelo com os servidores via ssh, para atualizar as variáveis presentes do modelo.

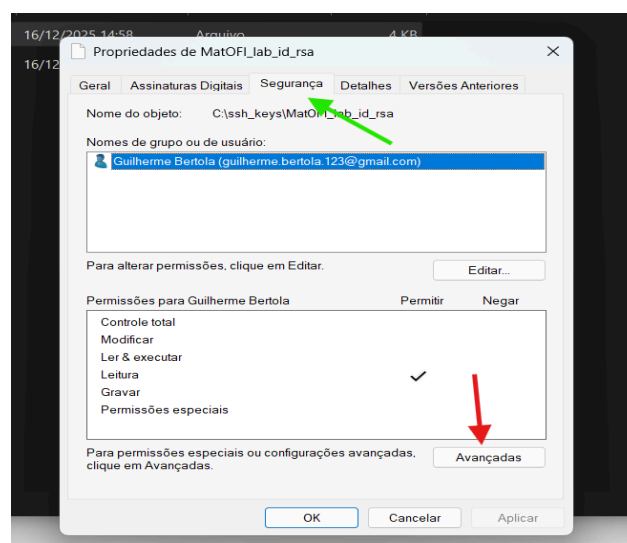
A configuração consiste em criar uma pasta nomeada “**ssh\_keys**” no diretório raiz do windows, para isso acesse o “**Explorador de Arquivos**”, localize o item “**Este computador**”, utilize o HD/SSD que possui o sistema operacional instalado, após isso adicione a pasta “**ssh\_keys**” conforme a imagem abaixo.



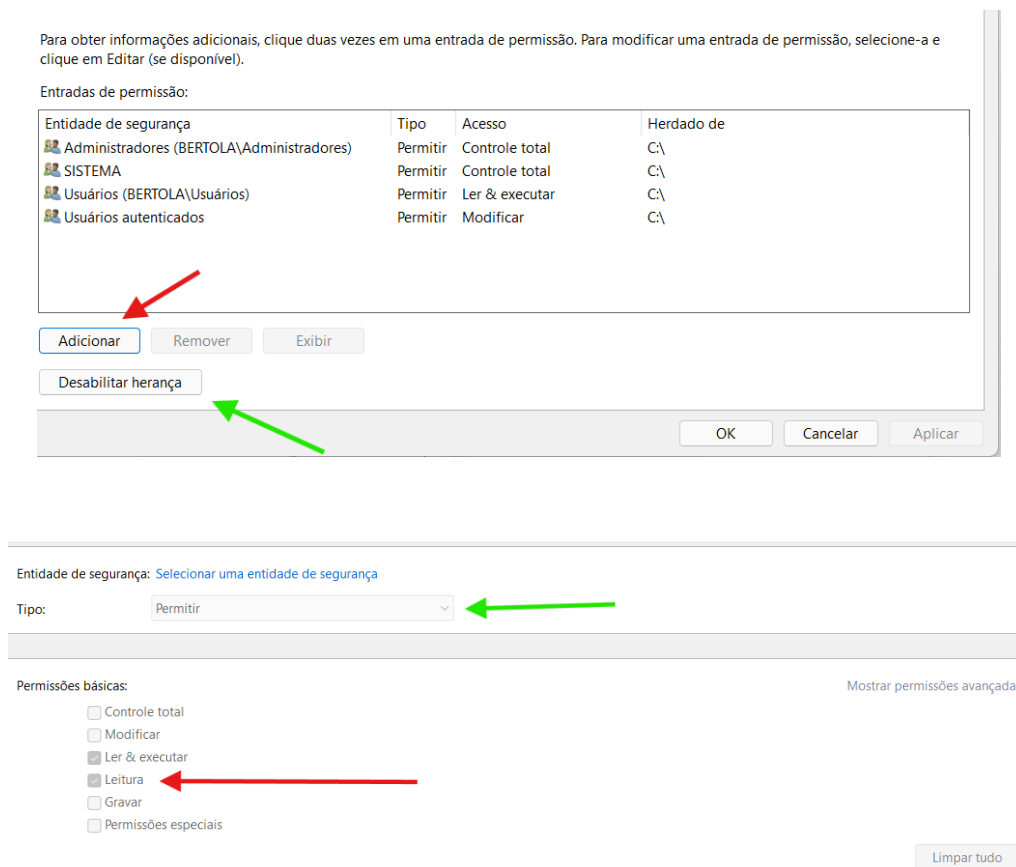
Após adicionar a pasta no modelo que foi instalado pelo link do github localize a pasta nomeada por “**Bertola’s Work**”, dentro do diretório terá outra pasta nomeada de “**ssh\_keys**” entrando nesta pasta copie os dois arquivos presentes chamados “**MatOFI\_lab\_id\_rsa**” e “**MatOFI\_lab\_id\_rsa.pub**” para dentro da pasta que foi criada anteriormente, em seguida clique com o botão direito no arquivo “**MatOFI\_lab\_id\_rsa**” já adicionado na pasta e procure por propriedades.



Em seguida, irá abrir uma janela de propriedades do arquivo e clique na aba de segurança e depois em avançado.



Após isso clique no botão desabilitar Herança e em seguida clique no botão adicionar, coloque seu usuário padrão da sua CPU e deixei tipo como “**Permitir**” e Permissão em “**Leitura**”.



Após desabilitar a herança de permissões e adicionar o usuário da sua CPU de “Ok” e aplique as modificações no arquivo. Concluindo assim a configuração de ambiente.

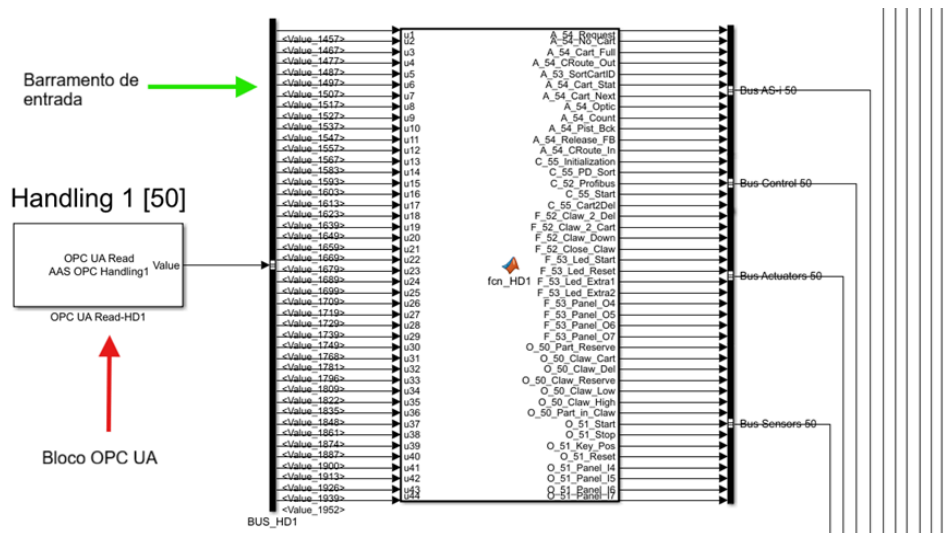
### 3. SCRIPT TAGS.M

O script nomeado como Tags.m, tem como função mapear todas as variáveis presentes no modelo feito no simulink, além disso esse script trabalha em conjunto com outro script “**Connect\_OPCUA.m**” feito para atualizar as variáveis presentes no próprio modelo, ou seja quando possui alguma adição, remoção ou alteração de variável.

#### 4. SCRIPT CONNECT\_OPCUA.M

Esse script é necessário no modelo, pois atualiza, deixa em ordem as variáveis e adiciona todas as variáveis que estão presentes no script “**Tags.m**” nos barramentos de entrada, estes barramentos são necessários para separar as variáveis que estão no bloco OPC UA de cada estação do modelo. Mas para o funcionamento adequado do script, antes de rodar o mesmo é necessário abrir o modelo no simulink e realizar o “**Procedimento para criação de múltiplas conexões remotas no MatLab/Simulink para servidores OPC UA**” e após a configuração com o modelo aberto deve-se rodar o script.

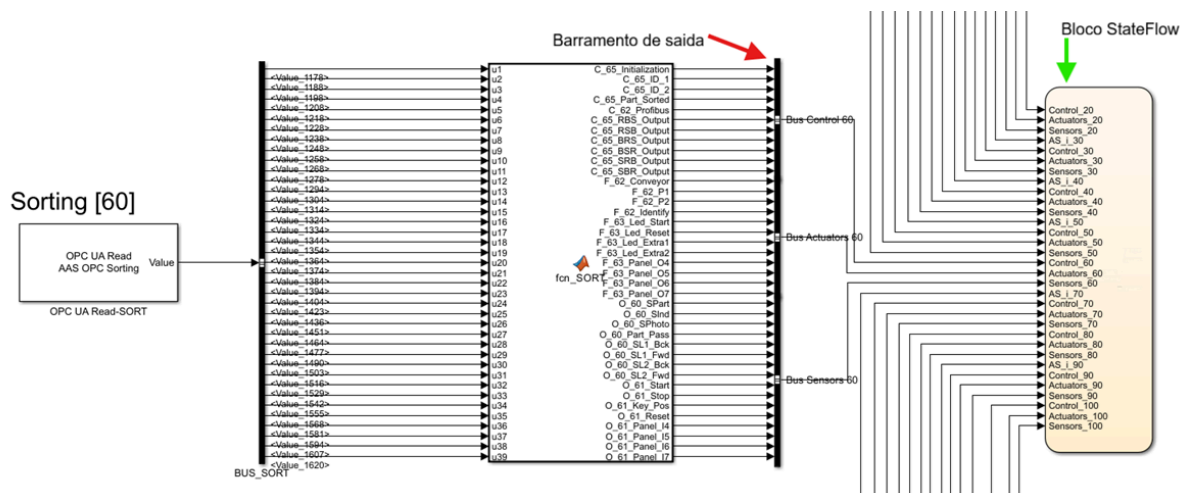
Segue a imagem do bloco OPC UA e os barramentos de entrada:



#### 5. TABELA BUS\_CONFIG.MAT

Esse arquivo .mat considerado uma “tabela” para o Matlab é responsável por configurar os barramentos de saídas, ou seja eles são necessários para configurar de qual tipo é cada variável (ex: boolean) e quais as variáveis presentes em cada barramento de saída, isso é importante para organizar as variáveis em grandes objetos para assim ser inseridos no bloco do stateflow, deixando o número de entradas do stateflow menores e mais organizados.

Segue a imagem dos barramentos de saída e o bloco do stateflow:



## 6. PROCEDIMENTO PARA CRIAÇÃO DE MÚLTIPLAS CONEXÕES REMOTAS NO MATLAB/SIMULINK PARA SERVIDORES OPC UA

Para realizar a conexão de múltiplos servidores OPC UA no MatLab de forma remota (onde o MatLab não está rodando na mesma máquina que o servidor) é necessário a configuração de basicamente 2 etapas. Após a configuração inicial será possível encontrar os servidores e configurá-los.

### 1 - Criação da lista de hosts no micro que rodará o MatLab

Localizar a pasta “C:\Windows\System32\drivers\etc”.

Dentro desta pasta, conterà um arquivo com o nome “hosts” (sem extensão).

Abrir esse arquivo (com permissão de administrador) e incluir a tabela de IP’s e Hostnames que serão utilizados para a conexão. No exemplo abaixo temos os IP’s e os Hostnames das TVBox que estão no Laboratório de Automação II.

200.145.26.202	TVBox-Master-20
200.145.26.213	TVBox-Robot-30
200.145.26.203	TVBox-Handling1-50
200.145.26.204	TVBox-Sorting-60
200.145.26.205	TVBox-Testing-70
200.145.26.206	TVBox-Distribution-80
200.145.26.207	TVBox-Handling2-90
200.145.26.208	TVBox-Processing-100

### 2 - Alteração do nome dos servidores nos destinos

Para realizar a alteração do nome dos servidores do OPC UA, no caso específico do Node-RED deve-se encontrar o arquivo “104-opcuaserver.js”.

Usar o localizador do Windows ou instalar o Samba no Linux para ter acesso a ferramenta de busca. Normalmente esse arquivo será encontrado na pasta “node\_modules/node-red-contrib-opcua/opcua/104-opcuaserver.js”

Uma vez que o arquivo foi encontrado, abrir ele com um editor de códigos (por exemplo o VSCode) e buscar pelo termo “applicationName”

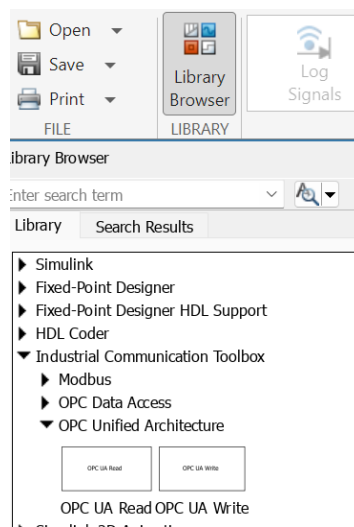
Ao encontrar a linha `applicationName: { text: "Node-RED OPCUA" }` basta mudar o texto que está entre as aspas duplas para um novo nome.

Salvar o arquivo e reiniciar o node-red.

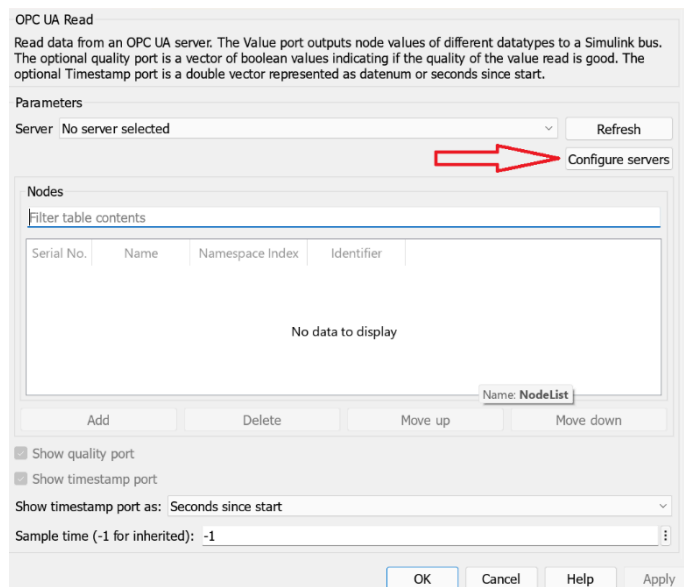
### 3 - Adição dos servidores no Simulink

Com a configuração inicial realizada, iniciar o MatLab e entrar no Simulink.

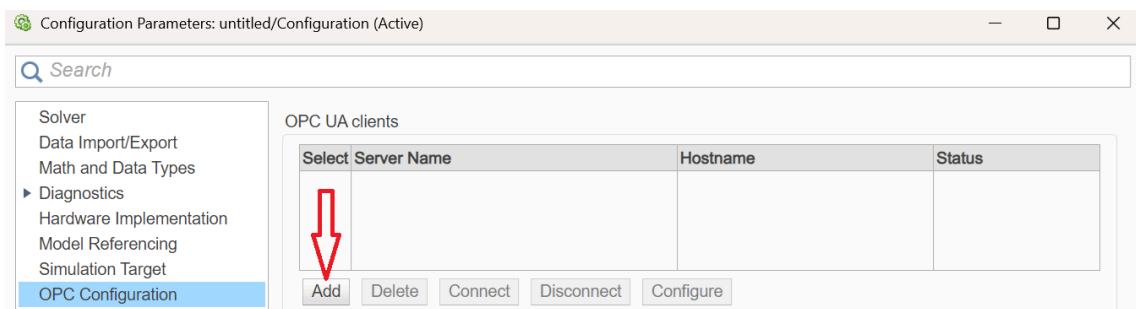
Clicar em Library Browser, depois em Industrial Communication Tools e por fim em OPC Unified Architecture para adicionar um dos blocos (de leitura ou de escrita).



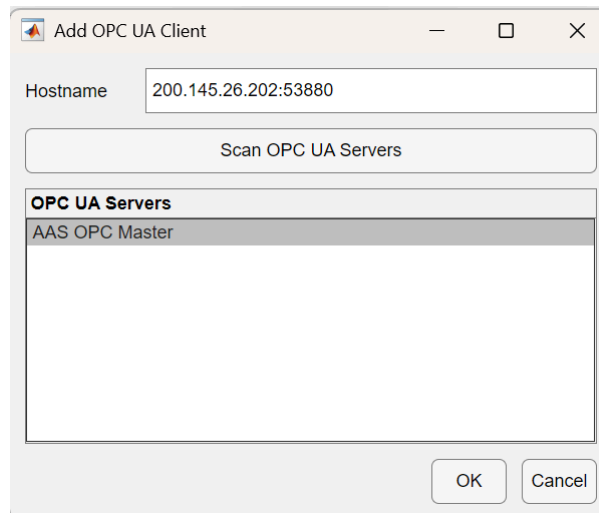
Uma vez que o módulo foi adicionado, clicar duas vezes no módulo e depois em “Configure Server”



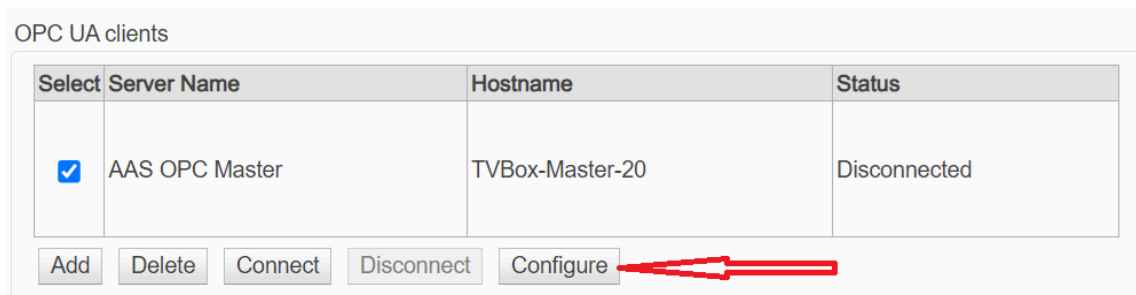
Na próxima tela, clicar em “ADD”



Na tela que abrir, na linha do Hostname colocar o IP e a porta que está sendo utilizada. No exemplo abaixo estou usando 200.145.26.202:53880. Clicar no botão “Scan OPC UA Servers” e o nome configurado no passo 2 deverá aparecer.

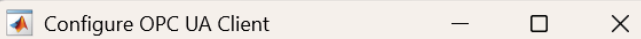


Após dar Ok, entrar nas configurações do Servidor.



Alterar as configurações de segurança para "None", o que fará com que o MatLab não peça certificado para o servidor.





Server Information

Name: AAS OPC Master

Hostname: TVBox-Master-20

Port: 53880

Endpoint URL: opc.tcp://TVBox-Master-20:53880/

Connection Information

Server State: <Not connected>

Status: Disconnected

Timeout: 10s

Select Security Policy

Select	Channel Security Policy	Message Security Mode
<input type="checkbox"/>	Basic128Rsa15	Sign
<input type="checkbox"/>	Basic128Rsa15	SignAndEncrypt
<input type="checkbox"/>	Basic256	Sign
<input type="checkbox"/>	Basic256	SignAndEncrypt
<input type="checkbox"/>	Basic256Sha256	Sign
<input type="checkbox"/>	Basic256Sha256	SignAndEncrypt
<input checked="" type="checkbox"/>	None	None

OK

Cancel

Após esse passo, basta clicar em “Connect” e escolher o usuário Anônimo.

Repetir o procedimento para os demais servidores que se deseja inserir.

OPC UA clients

Select	Server Name	Hostname	Status
<input type="checkbox"/>	AAS OPC Master	TVBox-Master-20	Connected
<input type="checkbox"/>	AAS OPC Handling1	TVBox-Handling1-50	Connected
<input type="checkbox"/>	AAS OPC Sorting	TVBox-Sorting-60	Connected
<input type="checkbox"/>	AAS OPC Testing	TVBox-Testing-70	Connected
<input type="checkbox"/>	AAS OPC Distribution	TVBox-Distribution-80	Connected
<input type="checkbox"/>	AAS OPC Handling2	TVBox-Handling2-90	Connected
<input type="checkbox"/>	AAS OPC Processing	TVBox-Processing-100	Connected

Add

Delete

Connect

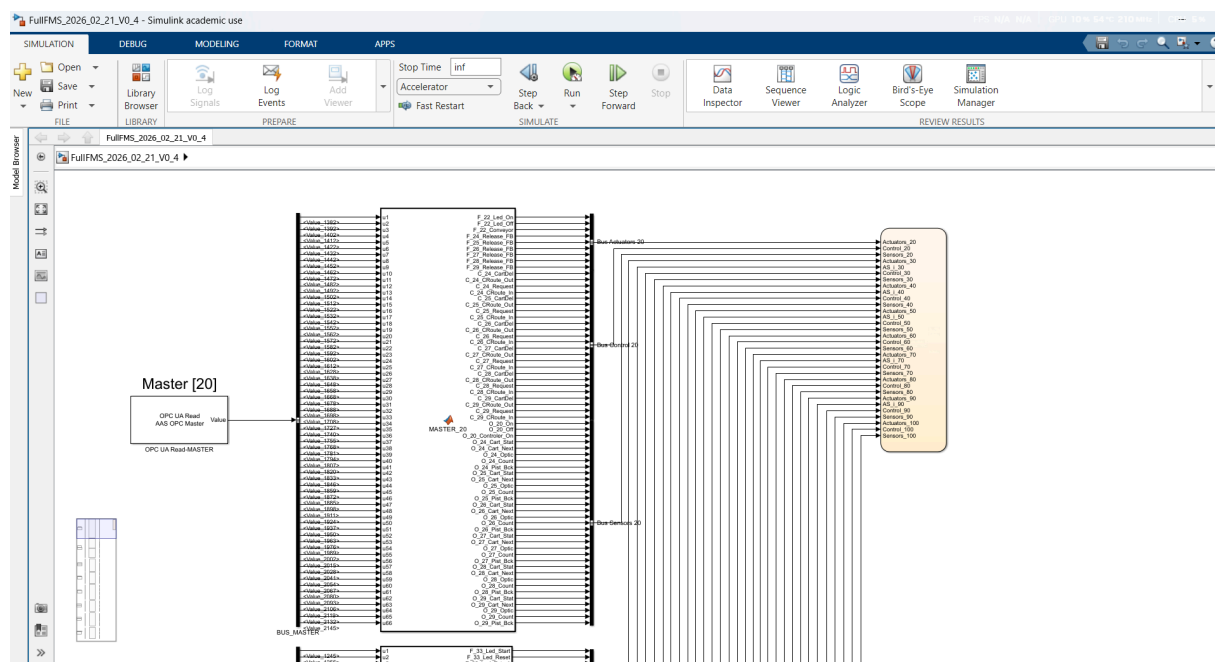
Disconnect

Configure

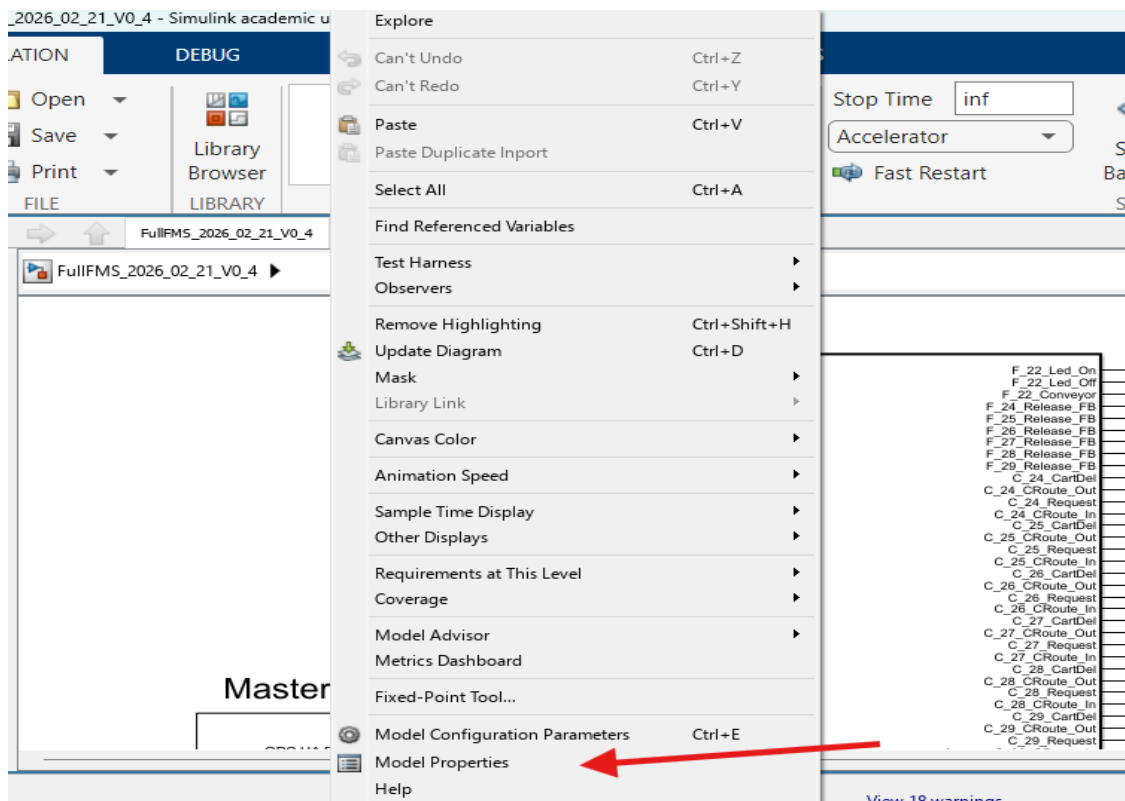
## 7. RENOMEAÇÃO DOS ARQUIVOS DO MODELO NO SIMULINK E DOS SEUS BLOCOS

Por possuir alguns scripts que proporcionam o funcionamento do modelo da melhor maneira possível, o modelo e script's/arquivos presentes criam dependências em seus códigos entre si, ou seja, o modelo quando iniciado a primeira vez “chama” o arquivo BUS\_CONFIG.mat para carregar os barramentos de saída das funções do matlab dentro do simulink, dando os tipo de cada variável para entrada no bloco do StateFlow. Porém, quando muda o nome do arquivo o modelo não consegue mais executar esse arquivo pois não acha mais o nome especificado no modelo que seria o BUS\_CONFIG.mat. Caso precise alterar o nome do arquivo é necessário a mudança do dentro do modelo também, seguindo os seguintes passos:

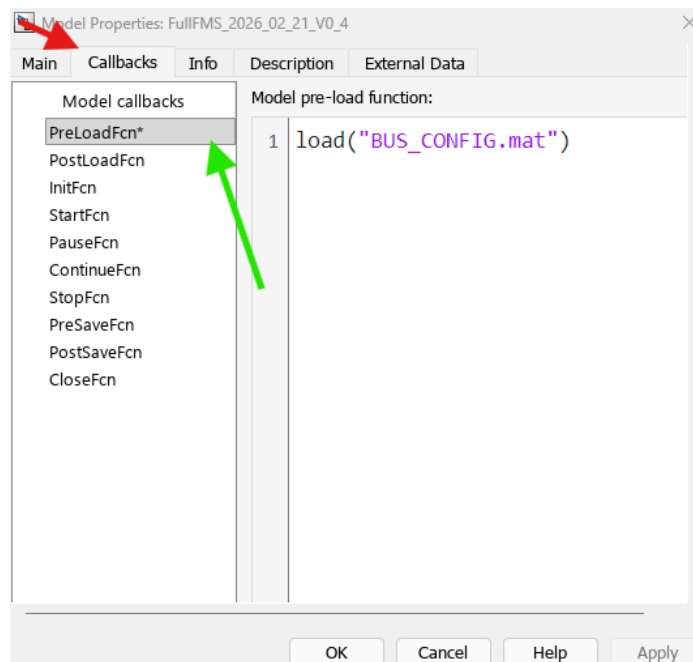
1 - Abra o modelo no simulink:



2 - Pressione com o botão direito do mouse e procure por Model Properties



3 - Abra o Model Properties e procure a aba Callbacks e em seguida clique em PreLoadFcn

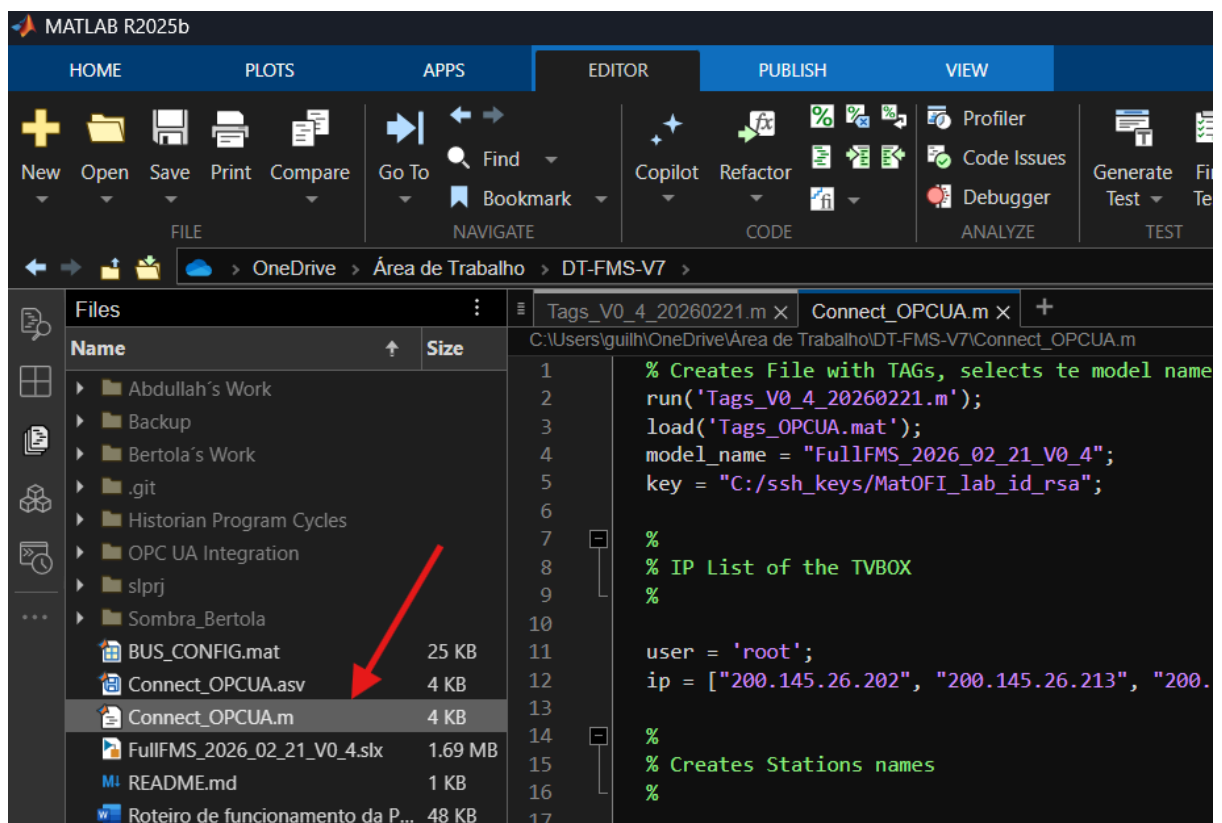


4 - Após o passo 3 mude o nome do conteúdo dentro de load com o novo nome do arquivo (ex: bus\_123.mat) e em seguida clique em Apply.

O script Connect\_OPCUA.m conforme já explicado anteriormente modifica e seleciona variáveis no modelo do simulink, sendo necessário antes abrir o próprio modelo antes de rodar o script.

Tecnicamente no ponto de vista de conexão com o modelo o script “chama” outro script (Tags.m) e em seguida reconhece o modelo do simulink para fazer as alterações necessárias que foram adicionadas e também analisa se a pasta “ssh\_keys” está presente no sistema operacional e se possui os arquivos necessários nela. Portanto, quando o for mudado o nome do modelo e do script Tags.m é necessário editar os nomes dentro do script Connect\_OPCUA.m. Seguindo os passos:

1 - Abra o script Connect\_OPCUA.m. através do matlab



2 - mude os nomes do conteúdo das linhas 2 e 4 para os novos nomes adicionados (ex: TagsNew.m e Modelo123), sendo a linha 2 o nome do script Tags.m e a linha 4 o nome do arquivo do modelo no simulink.

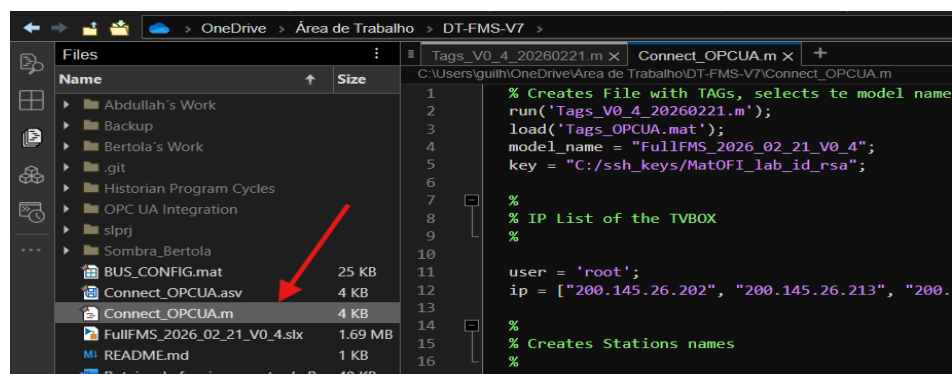
```
C:\Users\guihl\OneDrive\Área de Trabalho\DT-FMS-V7\Connect_OPCUA.m
1      % Creates File with TAGs, selects te model name and locate the SSH keys.
2      run('Tags_V0_4_20260221.m');
3      load('Tags_OPCUA.mat');
4      model_name = "FullFMS_2026_02_21_V0_4";
5      key = "C:/ssh_keys/MatOFI_lab_id_rsa";
6
7      %
8      % IP List of the TVBOX
9      %
10
11     user = 'root';
12     ip = ["200.145.26.202", "200.145.26.213", "200.145.26.213", "200.145.26.213"];
13
14     %
15     % Creates Stations names
16     %
17
```

3 - Após as devidas alterações aperte **Ctrl+s** salvando o script com o novo conteúdo.

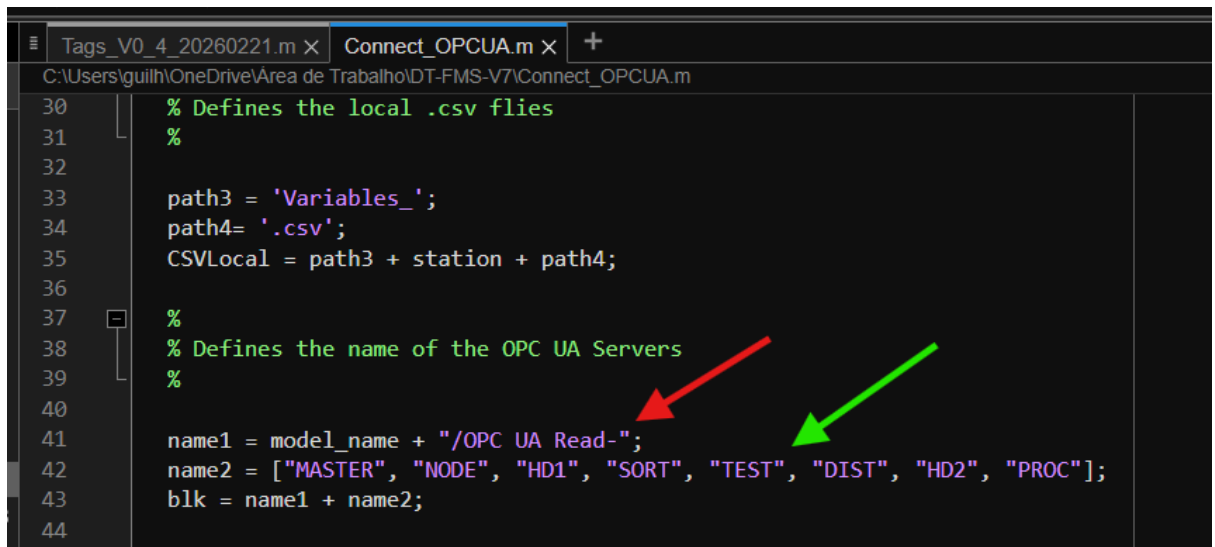
O script Connect\_OPCUA.m também analisa os blocos OPC UA Read dos modelos e os barramentos de entrada para organizar e separar as variáveis da conexão do OPC UA, o script faz isso através dos nomes dados aos blocos OPC e aos barramentos de entrada.

Caso seja necessário mudar os nomes dos blocos OPC UA ou até mesmo dos barramentos de entradas conectados aos blocos OPC UA é preciso mudar os nomes já tabelados no script, seguindo os passos a seguir:

1 - Abra o script Connect\_OPCUA.m. através do matlab.

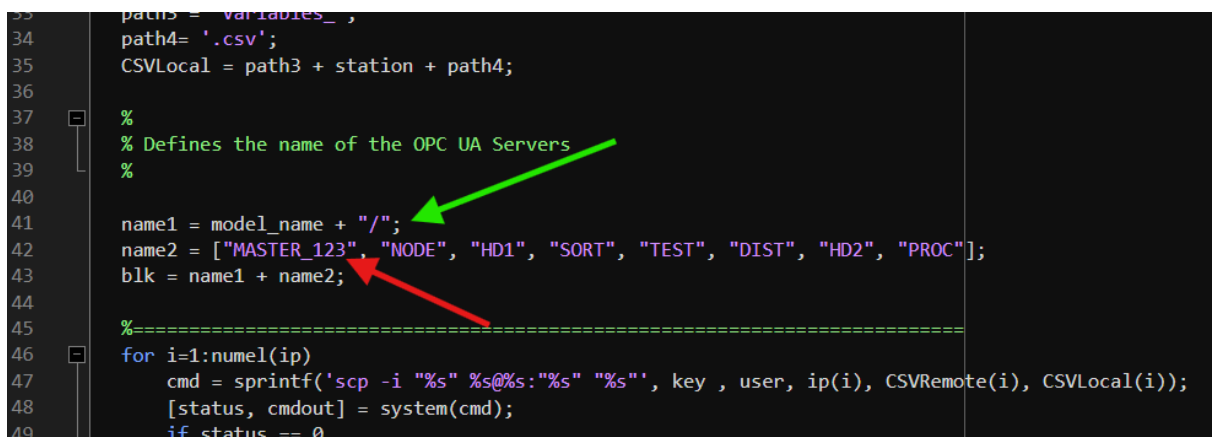


## 2 - Procure pelas linhas 41, 42 e 43 caso mude o nome dos blocos OPC UA



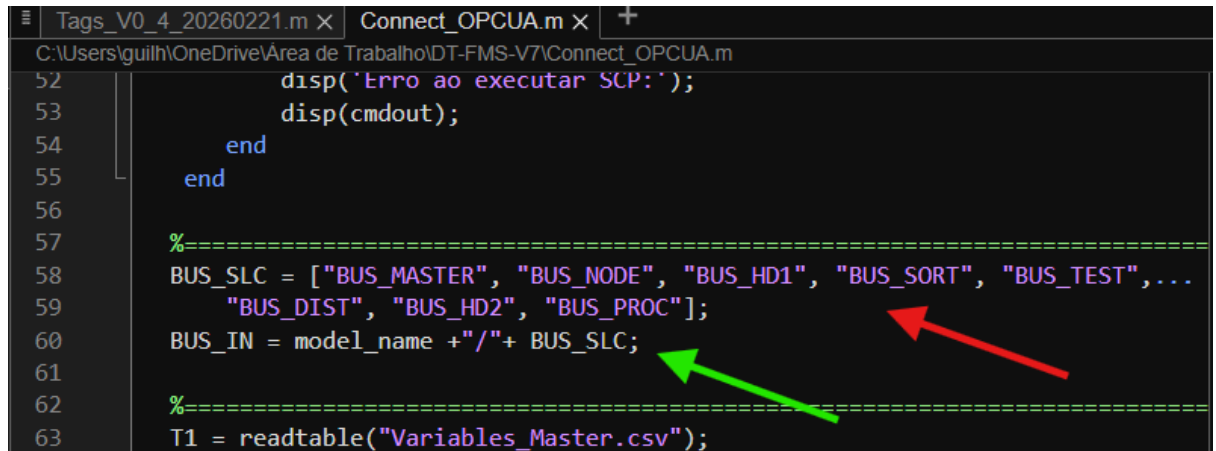
```
30 % Defines the local .csv files
31 %
32
33 path3 = 'Variables_';
34 path4 = '.csv';
35 CSVLocal = path3 + station + path4;
36
37 %
38 % Defines the name of the OPC UA Servers
39 %
40
41 name1 = model_name + "/OPC UA Read-";
42 name2 = ["MASTER", "NODE", "HD1", "SORT", "TEST", "DIST", "HD2", "PROC"];
43 blk = name1 + name2;
44
```

A estrutura de busca dos blocos é realizada por endereço, ou seja o endereço que o matlab utiliza para modificar modelos no simulink é por “**nome\_modelo/nome\_do\_bloco**”, para caso mude o nome no bloco do modelo é necessário mudar o nome do bloco no script tanto na linha 41 na parte entre as **aspas duplas** e na linha 42 referente a ordem do bloco que foi mudado no modelo (ex: o primeiro bloco OPC UA, se refere ao primeiro índice do array da linha 42 “MASTER”), seguindo esse código padrão geramos a seguinte string (“nome\_modelo/OPC UA Read-MASTER”), caso o novo nome por apenas “MASTER\_123” é preciso tirar “OPC UA Read-” da linha 42 deixando apenas “/” e mudar na linha 43 o conteúdo que está entre aspas duplas do primeiro índice que se refere ao bloco do master para “MASTER\_123”, seguindo essa lógica para os demais blocos caso necessário. Ficando seguinte a imagem:



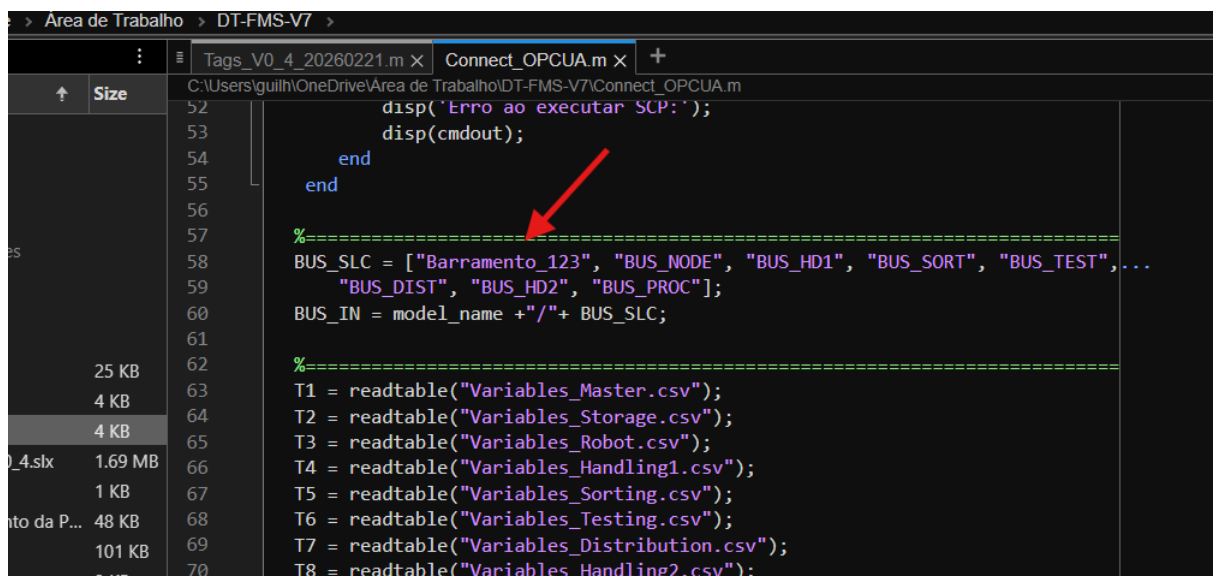
```
33 path3 = 'Variables_';
34 path4 = '.csv';
35 CSVLocal = path3 + station + path4;
36
37 %
38 % Defines the name of the OPC UA Servers
39 %
40
41 name1 = model_name + "/";
42 name2 = ["MASTER_123", "NODE", "HD1", "SORT", "TEST", "DIST", "HD2", "PROC"];
43 blk = name1 + name2;
44
45 %=====
46 for i=1:numel(ip)
47     cmd = sprintf('scp -i "%s" %s@%s:"%s" "%s"', key, user, ip(i), CSVRemote(i), CSVLocal(i));
48     [status, cmdout] = system(cmd);
49     if status == 0
```

3 - Caso for necessário mudar os nomes dos barramentos de entrada, procure as linhas 58, 59 e 60.



```
52     disp('Erro ao executar SCP:');
53     disp(cmdout);
54     end
55     end
56
57     %=====
58     BUS_SLC = ["BUS_MASTER", "BUS_NODE", "BUS_HD1", "BUS_SORT", "BUS_TEST",...
59              "BUS_DIST", "BUS_HD2", "BUS_PROC"];
60     BUS_IN = model_name + "/" + BUS_SLC;
61
62     %=====
63     T1 = readtable("Variables_Master.csv");
```

Portanto, se precisar mudar o nome dos barramentos, certifique qual barramento foi modificado e verifique qual é o seu índice na linha 58 e 59 e mude o conteúdo **entre as aspas duplas**, lembrando que os barramentos estão em ordem com os blocos OPC UA de forma de manter a organização. Por exemplo, se quiser mudar o nome do barramento do master para **Barramento\_123**, apenas precisa modificar na linha 58 o conteúdo do respectivo índice do master, conforme a imagem abaixo.



```
> Area de Trabalho > DT-FMS-V7 >
Tags_V0_4_20260221.m x Connect_OPCUA.m x +
C:\Users\guihl\OneDrive\Área de Trabalho\DT-FMS-V7\Connect_OPCUA.m
52     disp('Erro ao executar SCP:');
53     disp(cmdout);
54     end
55     end
56
57     %=====
58     BUS_SLC = ["Barramento_123", "BUS_NODE", "BUS_HD1", "BUS_SORT", "BUS_TEST",...
59              "BUS_DIST", "BUS_HD2", "BUS_PROC"];
60     BUS_IN = model_name + "/" + BUS_SLC;
61
62     %=====
63     T1 = readtable("Variables_Master.csv");
64     T2 = readtable("Variables_Storage.csv");
65     T3 = readtable("Variables_Robot.csv");
66     T4 = readtable("Variables_Handling1.csv");
67     T5 = readtable("Variables_Sorting.csv");
68     T6 = readtable("Variables_Testing.csv");
69     T7 = readtable("Variables_Distribution.csv");
70     T8 = readtable("Variables_Handling2.csv");
```

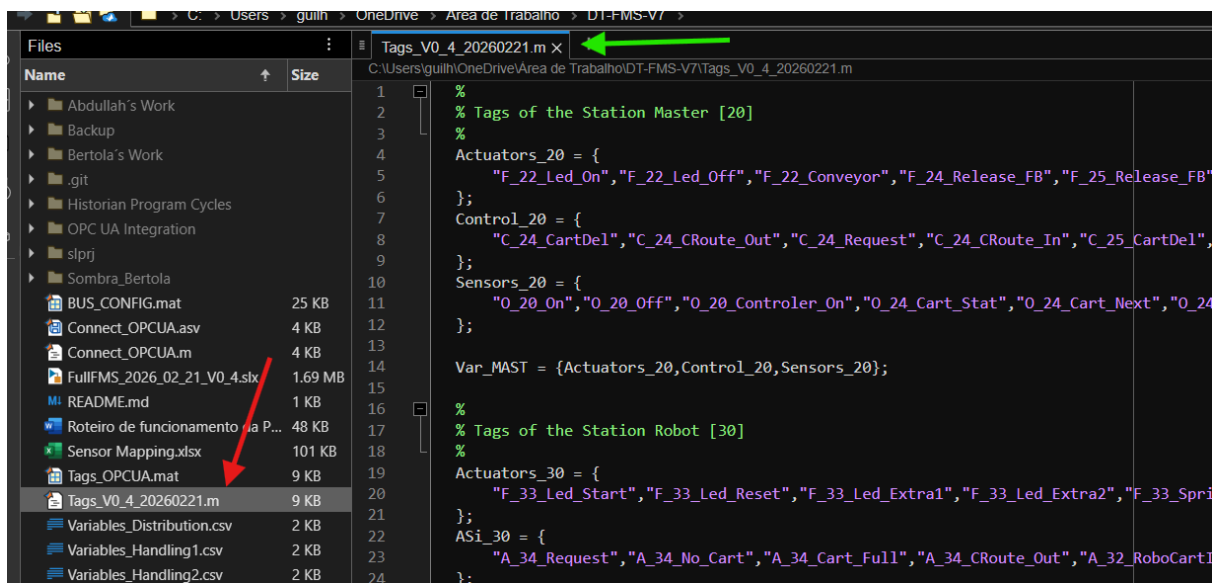
4 - Após mudar e modificar todos os nomes necessários no modelo do simulink e no script conforme feito no modelo do simulink, basta apertar **Ctrl+s** para salvar o script modificado.

## 8. ADIÇÃO, ALTERAÇÃO E REMOÇÃO DE VARIÁVEIS

Caso haja a necessidade de adições, alterações e remoções nas variáveis presentes no modelo é preciso modificar quatro coisas dentro do projeto em geral, sendo essas coisas o **script Tags.m**, a **tabela BUS\_CONFIG.mat** e **bloco de função do matlab** dentro do modelo no simulink.

### Modificando Tags.m

1 - Abra o script Tags.m. através do matlab.



2 - Em seguida procure a bancada que deseja fazer a alteração pela sua numeração (ex: master → 20), identifique qual seção a variável que deseja adicionar ou remover/alterar deve estar e modifique o conteúdo desta seção. Caso for adicionar variáveis coloque na ordem necessária exemplo “**var\_1**”, “**var\_2**” e etc..

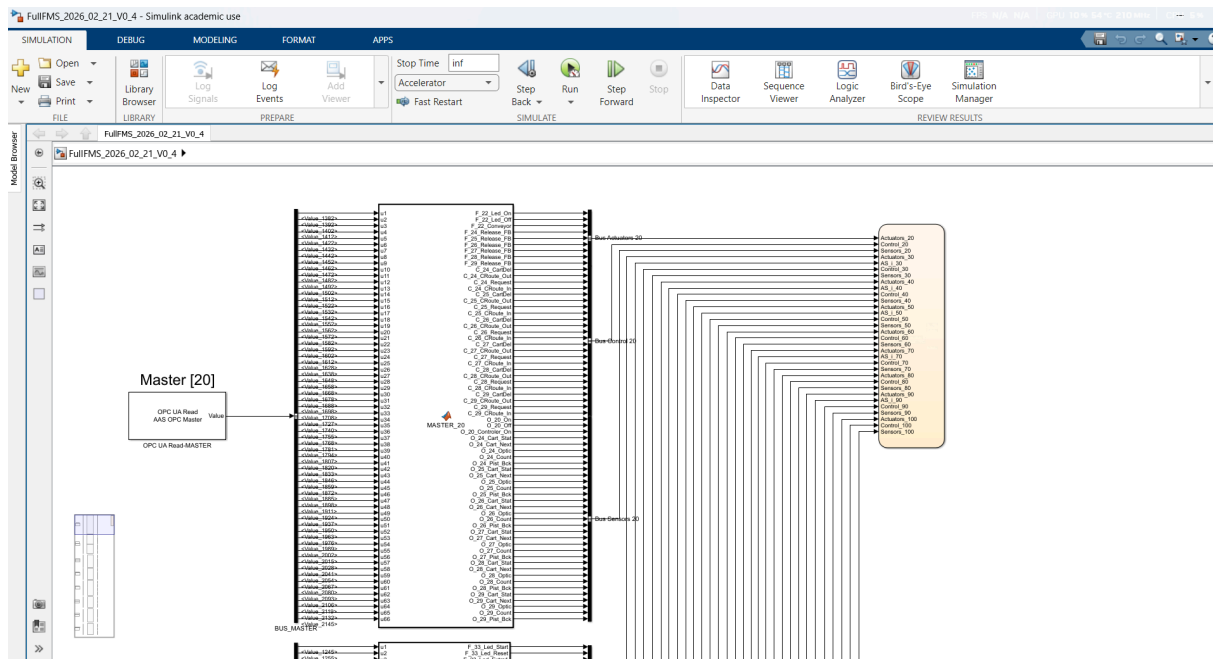
Caso for alterar/remover a variável apenas modifique o conteúdo entre as aspas ou até mesmo remova as aspas e o conteúdo dentro removendo a variável.

3 - Após as devidas alterações aperte **Ctrl+s** salvando o script com o novo conteúdo.

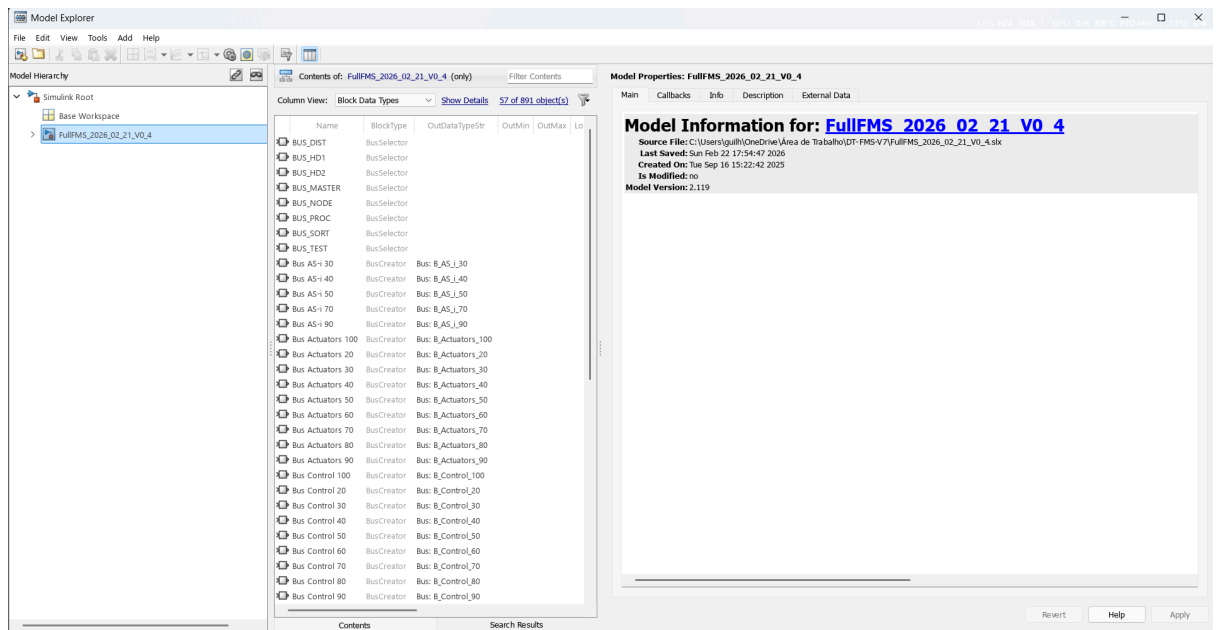
### Modificando a tabela BUS\_CONFIG.mat



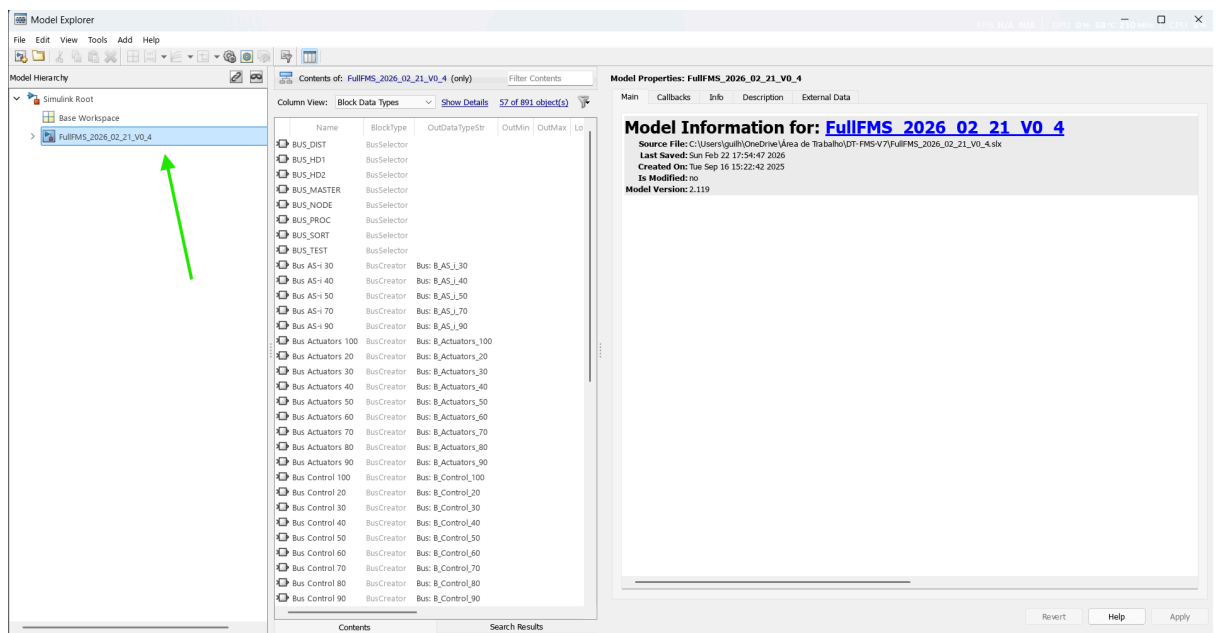
## 1 - Abra o modelo no simulink.



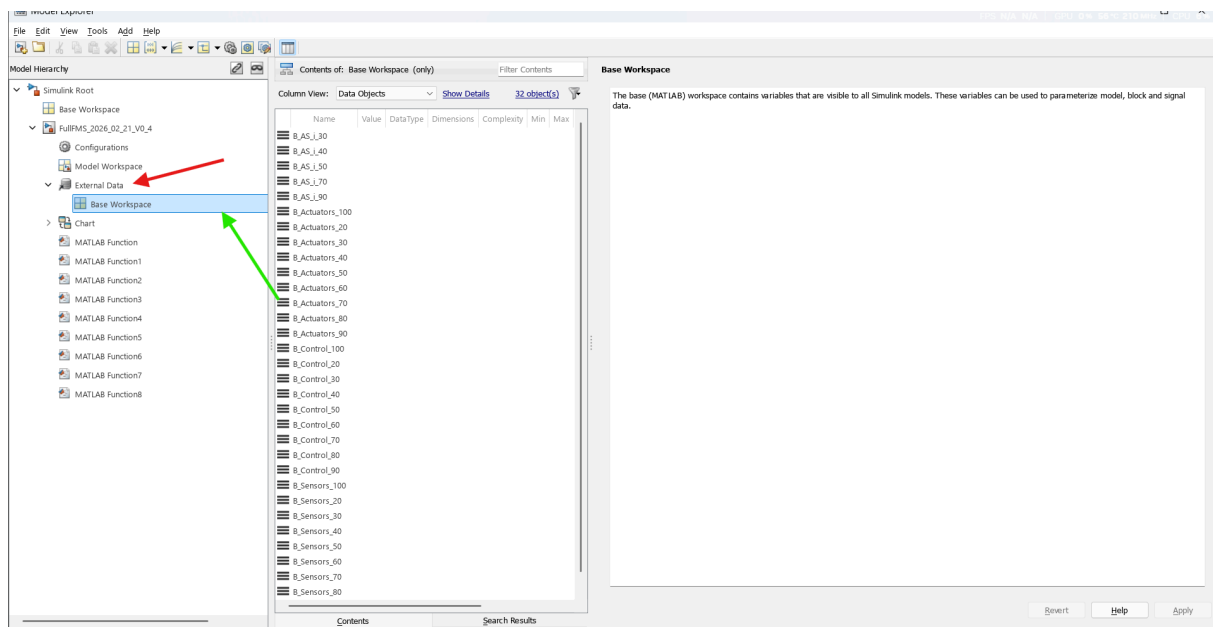
## 2 - aperte **Ctrl + h**, abrindo o Model Explorer



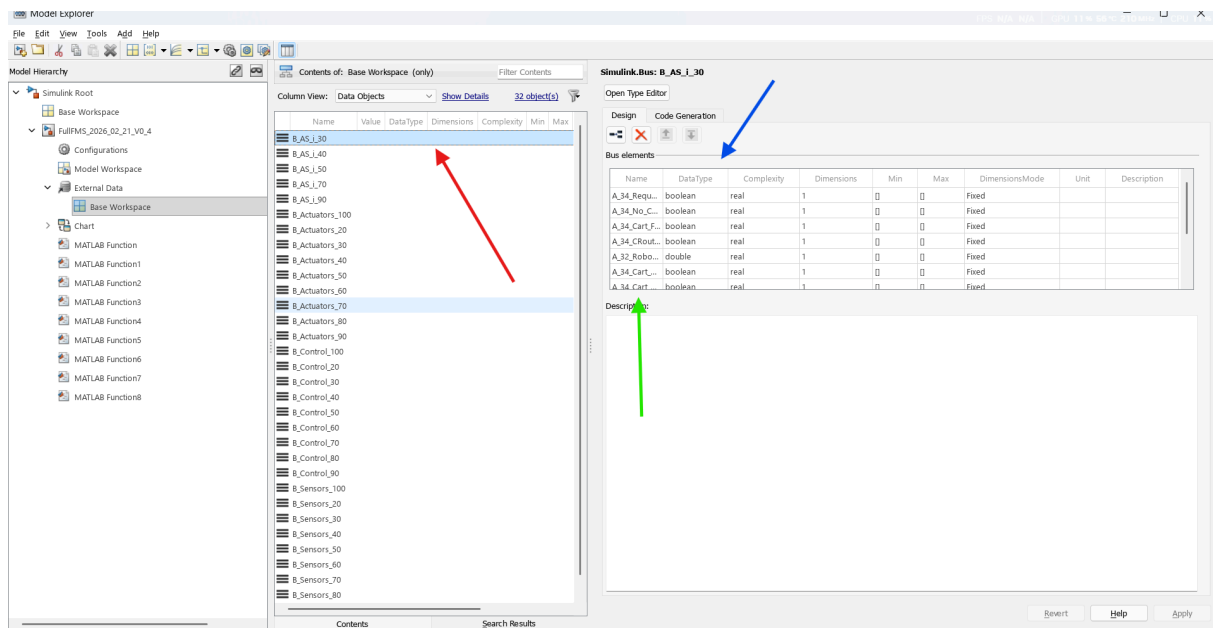
### 3 - após abrir clique no nome do seu modelo



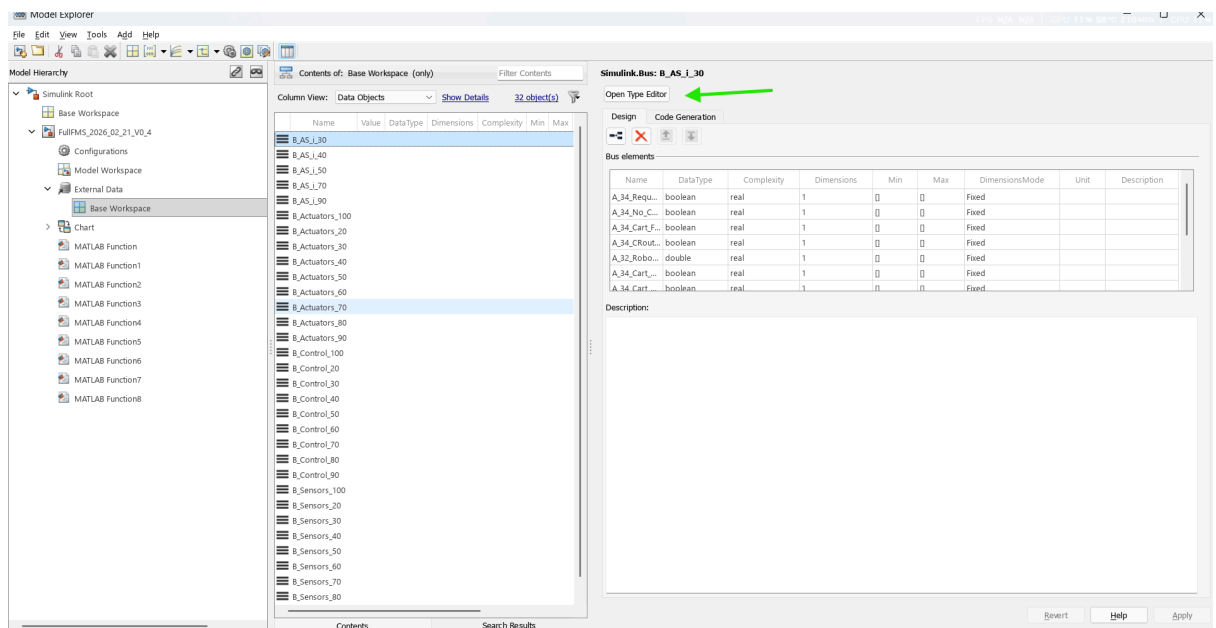
### 4 - Em seguida clique no External Data e em seguida em Base Workspace



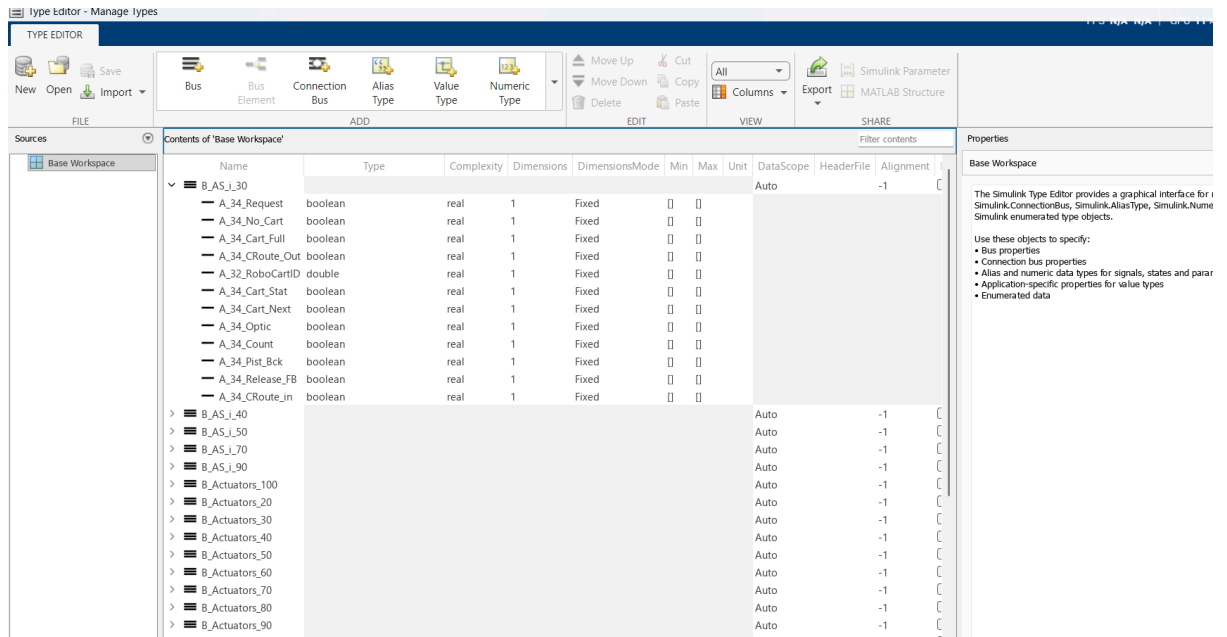
### 5 - procure em qual barramento a variável deve ser adicionada, alterada ou removida e clique no barramento, abrindo a lista de todas as variáveis nesse barramento e o tipo da variável.



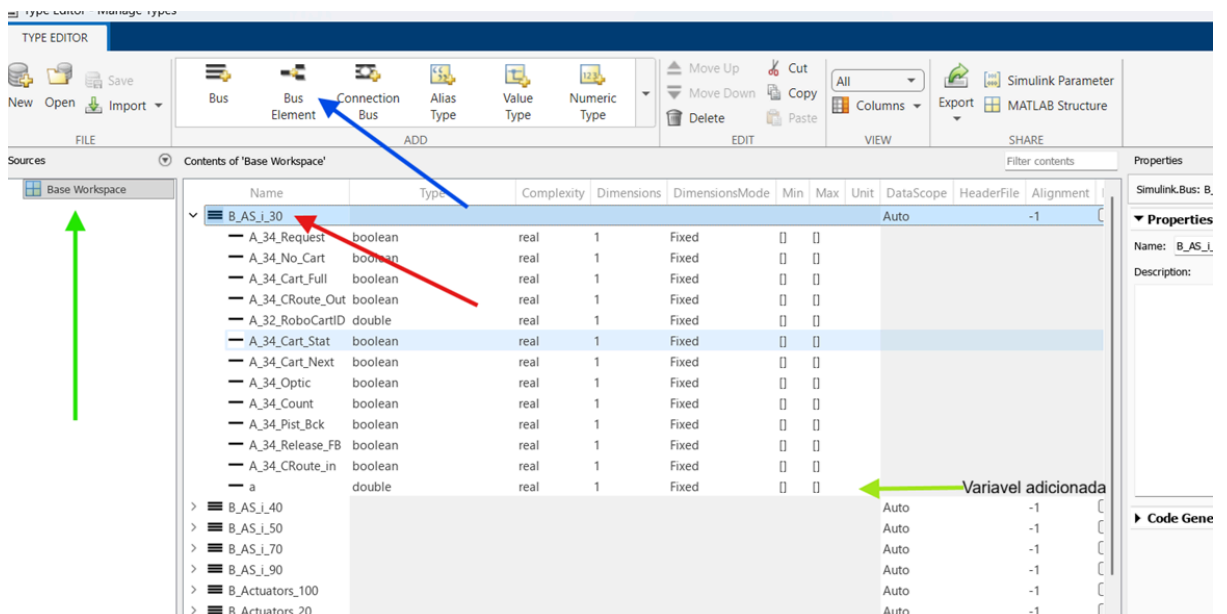
## 6 - Clique após selecionar o barramento o botão **Open Type Editor**



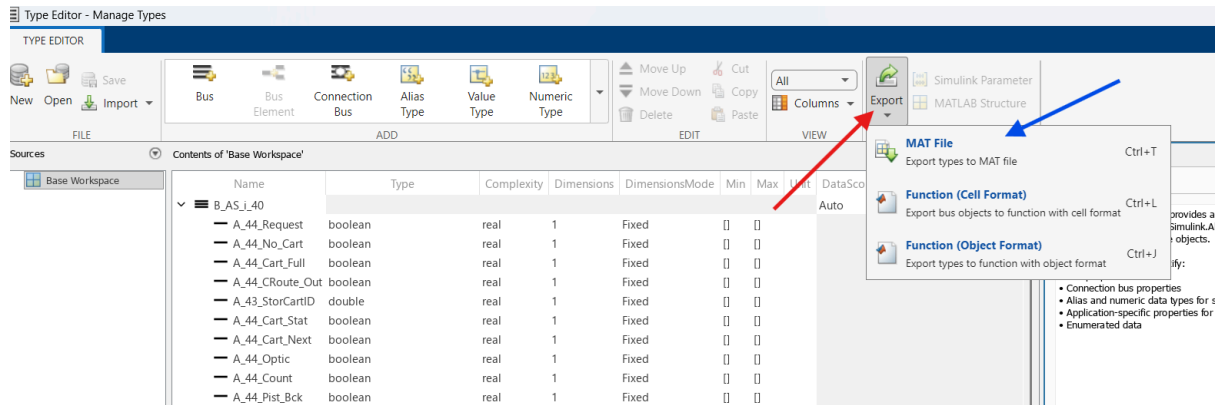
7 - Em seguida procure pelo barramento selecione, fazendo abrir todas as variáveis que estão configuradas naquele barramento selecionado e faça a modificação necessária.



8 - Selecione a variável desejada e modifique seu nome ou a remova e para adicionar mais variável clique no botão Bus element com o barramento desejado selecionado e caso necessário mudar a ordem das variáveis selecione a variável que deseja mudar a sua ordem e arraste até o local desejado no barramento. **OBS: não esqueça de colocar o nome na variável nova e seu tipo de variável (ex: boolean).**



9 - Para salvar a nova modificação clique em **Export**, selecione MAT File e sobrescreva o BUS\_CONFIG.mat da pasta do projeto.

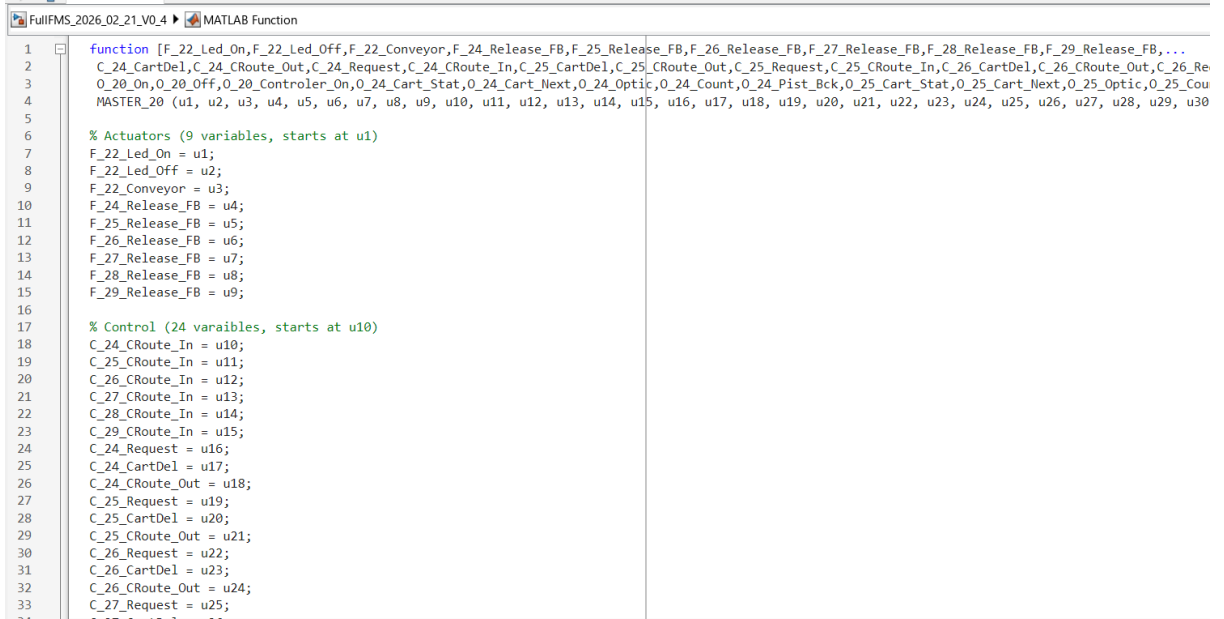


## Modificando o bloco de função do Matlab

1 - Abra o modelo no simulink.



2 - Clique no bloco de função desejado para entrar no seu ambiente de código.



```
1 function [F_22_Led_On,F_22_Led_Off,F_22_Conveyor,F_24_Release_FB,F_25_Release_FB,F_26_Release_FB,F_27_Release_FB,F_28_Release_FB,F_29_Release_FB,...
2 C_24_CartDel,C_24_CRoute_Out,C_24_Request,C_24_CRoute_In,C_25_CartDel,C_25_CRoute_Out,C_25_Request,C_25_CRoute_In,C_26_CartDel,C_26_CRoute_Out,C_26_Re
3 O_20_On,O_20_Off,O_20_Controller_On,O_24_Cart_Stat,O_24_Cart_Next,O_24_Optic,O_24_Count,O_24_Pist_Bck,O_25_Cart_Stat,O_25_Cart_Next,O_25_Optic,O_25_Cou
4 MASTER_20 (u1, u2, u3, u4, u5, u6, u7, u8, u9, u10, u11, u12, u13, u14, u15, u16, u17, u18, u19, u20, u21, u22, u23, u24, u25, u26, u27, u28, u29, u30
5
6 % Actuators (9 variables, starts at u1)
7 F_22_Led_On = u1;
8 F_22_Led_Off = u2;
9 F_22_Conveyor = u3;
10 F_24_Release_FB = u4;
11 F_25_Release_FB = u5;
12 F_26_Release_FB = u6;
13 F_27_Release_FB = u7;
14 F_28_Release_FB = u8;
15 F_29_Release_FB = u9;
16
17 % Control (24 variables, starts at u10)
18 C_24_CRoute_In = u10;
19 C_25_CRoute_In = u11;
20 C_26_CRoute_In = u12;
21 C_27_CRoute_In = u13;
22 C_28_CRoute_In = u14;
23 C_29_CRoute_In = u15;
24 C_24_Request = u16;
25 C_24_CartDel = u17;
26 C_24_CRoute_Out = u18;
27 C_25_Request = u19;
28 C_25_CartDel = u20;
29 C_25_CRoute_Out = u21;
30 C_26_Request = u22;
31 C_26_CartDel = u23;
32 C_26_CRoute_Out = u24;
33 C_27_Request = u25;
```

3 - Após isso adicione a variável necessária seguindo a ordem necessária, modifique alguma variável ou até mesmo remova as variáveis. Não esquecer de aumentar o número de parâmetros de entrada da função e o número de saídas da própria função quando adicionar variáveis ou diminuir os parâmetros de entradas e o número de saídas da função quando for remover variáveis.

4 - Após as devidas alterações aperte **Ctrl+s** salvando o modelo com o novo conteúdo.