

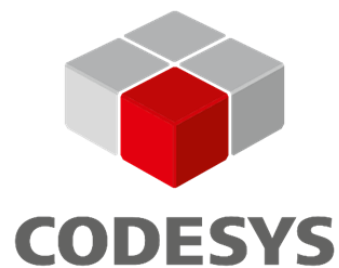
COMO IMPLEMENTAR UM SISTEMA RFID COM UM CLP WAGO E UMA ANTENA IO-LINK BALLUFF

Introdução

Este documento descreve o processo de implementação de um sistema RFID utilizando um CLP Wago e uma antena IO-Link da Balluff, com todo o desenvolvimento sendo realizado no software de programação CODESYS. Ao final desta nota de aplicação, você será capaz de realizar a leitura e escrita de tags, o que permitirá implementar diversas funcionalidades, como controle de acesso, gestão de inventário, rastreamento de produtos e suprimentos, entre outras.

Material Utilizado

- CLP PFC 200 com IP Estático configurado
- Remota IO-Link BNI00HL
- Antena IO-Link BIS01E2
- CODESYS V3.5 SP19 Patch 2



1 – Download dos arquivos necessários para o projeto

Antes de começar a implementação do sistema RFID, é fundamental baixar o descritivo Profinet do mestre IO-Link e a biblioteca com o bloco de operação da antena. Para isso, acesse o site da [Balluff](https://www.balluff.com), digite na barra de pesquisa o código da remota, nesse caso, BNI00HL, abra a seção de arquivos e baixe o item nomeado *GSDML*.

BALLUFF Produtos Setores e Soluções Serviços Empresa Contato & Serviço

Início > Produtos > Tecnologia de rede industrial > Módulos do distribuidor > Módulos de rede para Profinet > BNI00HL

BNI00HL

acessível
Módulos de rede para Profinet

innovating automation

Código de digitação:
BNI PG3-508-0C5-Z015

Adicionar à comparação Adicionar aos favoritos

Para preços individuais, por favor contacte-nos.
[INICIAR SESSÃO](#)

1 Adicionar ao carrinho
Comprar agora
Criar orçamento

Data de entrega: [RECUPERAR DADOS](#) Disponibilidade: [RECUPERAR DADOS](#)

Para o configurador RFID
Alterar configuração

Ilustração similar

BALLUFF Produtos Setores e Soluções Serviços Empresa Contato & Serviço

Características principais Acessórios recomendados **Arquivos** Vídeos Classificações Modelo CAD/CAE

Documentação técnica

Ficha técnica
Portuguese

Certificados MTTF:
PDF Download

Descarte
PDF Download

Software

Arquivo de descrição do aparelho Profinet (GSDML)
GSDML BNI PG3-508-0C5-Z015

Documentação do produto

Certificação
UL E319845-20120713 I23M99 (Inglês)
PNO Z13816 D24C27 (Inglês)

Configuration guide
BNI PG3-508-0C5_015 (Alemão, Inglês)

Em seguida, acesse a página da antena da mesma forma, siga para o menu de Arquivos e selecione o item *BIS FB TC* na lista *Bloco de função (FB)*.

BIS01E2

Cabeças de leitura/escrita HF (13,56 kHz) com processador integrado

innovating automation

Código de digitação:
BIS M-4A9-082-401-07-PU1-0,3

Adicionar à comparação Adicionar aos favoritos



Ilustração similar

Para preços individuais, por favor contacte-nos.

INICIAR SESSÃO

1 Adicionar ao carrinho

Comprar agora

Criar orçamento

ACTIVAR A NOTIFICAÇÃO DE DISPONIBILIDADE

Data de entrega

RECUPERAR DADOS

Disponibilidade

RECUPERAR DADOS

Para o configurador RFID

Alterar configuração



Características principais Acessórios recomendados Arquivos Classificações Modelo CAD/CAE



Documentação técnica

- Ficha técnica
- Portuguese
- Dados de medição
- PDF Download
- Certificados MTTF
- PDF Download
- Descarte
- PDF Download

Documentação do produto

- Anexo
- SAFETY INSTRUCTIONS
- BIS M-4A9.-.082-401-07-...
(Alemão, Inglês, Francês)
- Certificação
- UL E227256-D1003-1-A3-C0-UL G21M99
(Inglês)
- ECOLAB_BAUREIHE_BIS J20M99
(Inglês)
- FCC_2AGZY-BFIDM33 E21M99
(Inglês)



(Alemão)

BASICS RFID
(Inglês)

Manual de instruções

BIS M-4...082-401-07...

Software

Bloco de função (FB)

BIS FB PROFIBUS S7 BISM IO-LINK

BIS FB TIA

FB IO-LINK CM DEMAND DATA

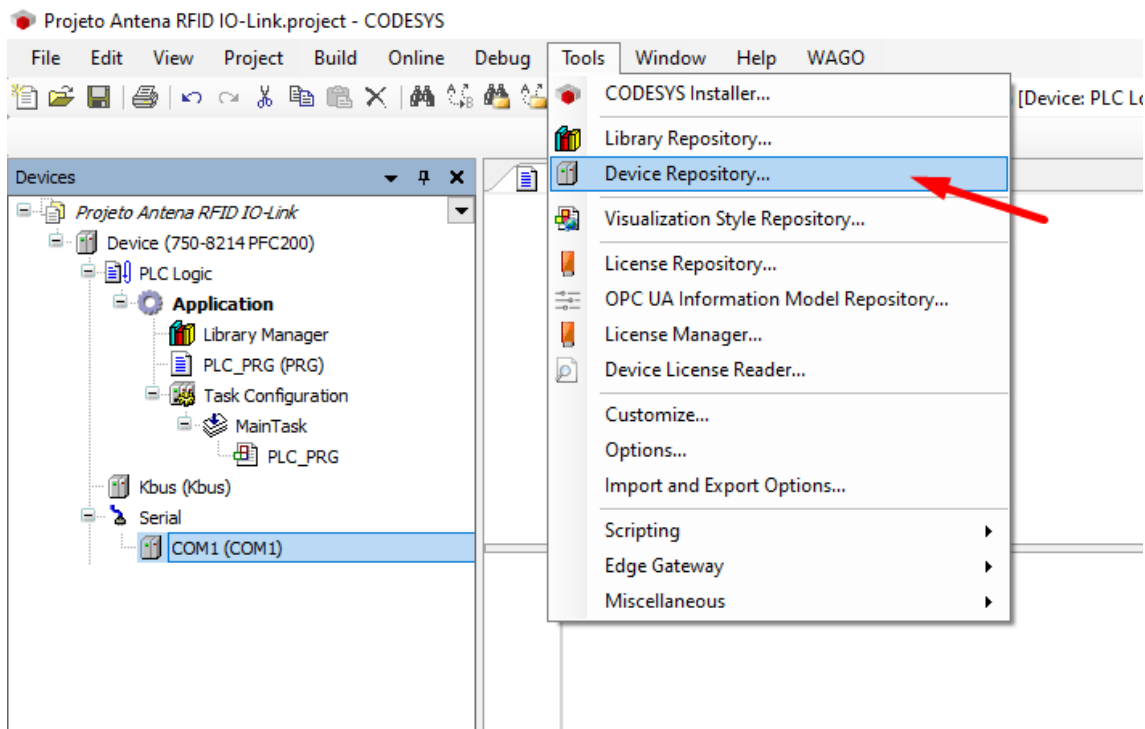
BIS FB TC

AOI BIS M-4Ax

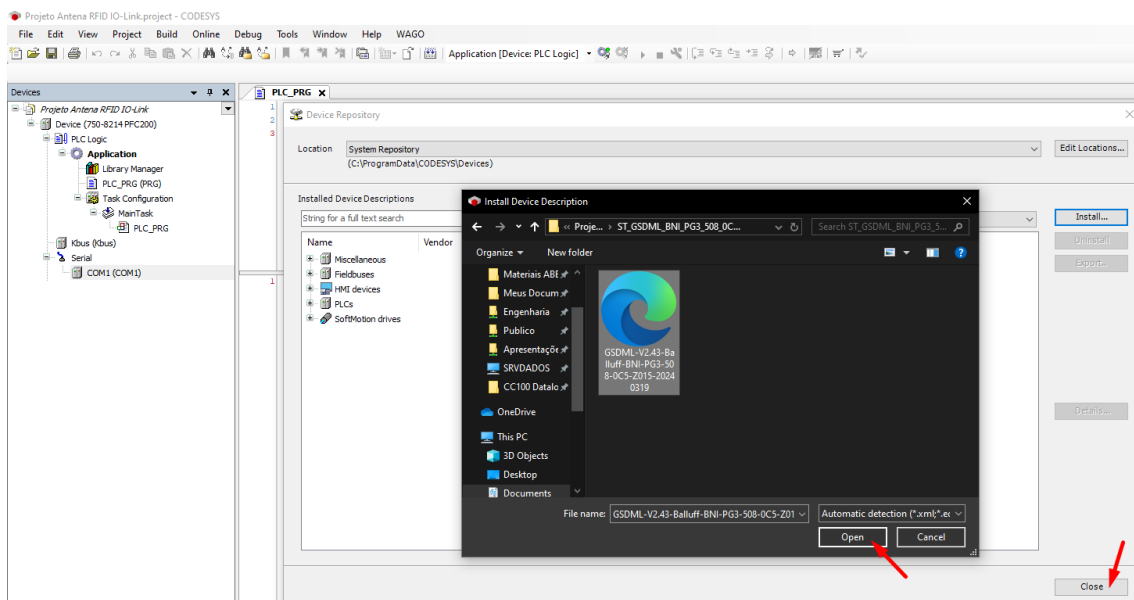
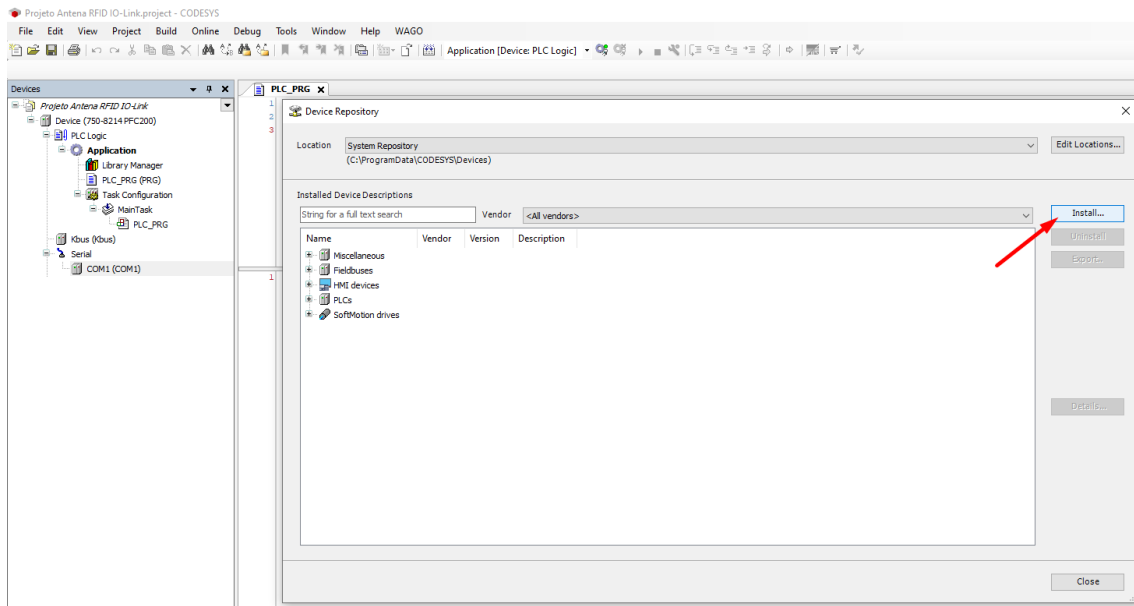
Terminado o download desses itens, realize a extração de ambos para uma pasta de sua preferência.

2 – Adicionando a remota IO-Link ao projeto

Nesse passo, iremos instalar o descritivo Profinet do mestre IO-Link na lista de dispositivos do Codesys. Dessa forma, abra seu projeto e acesse *Tools > Device Repository*.



Uma nova janela será aberta, clique em *Install* e adicione o arquivo GSDML baixado anteriormente.



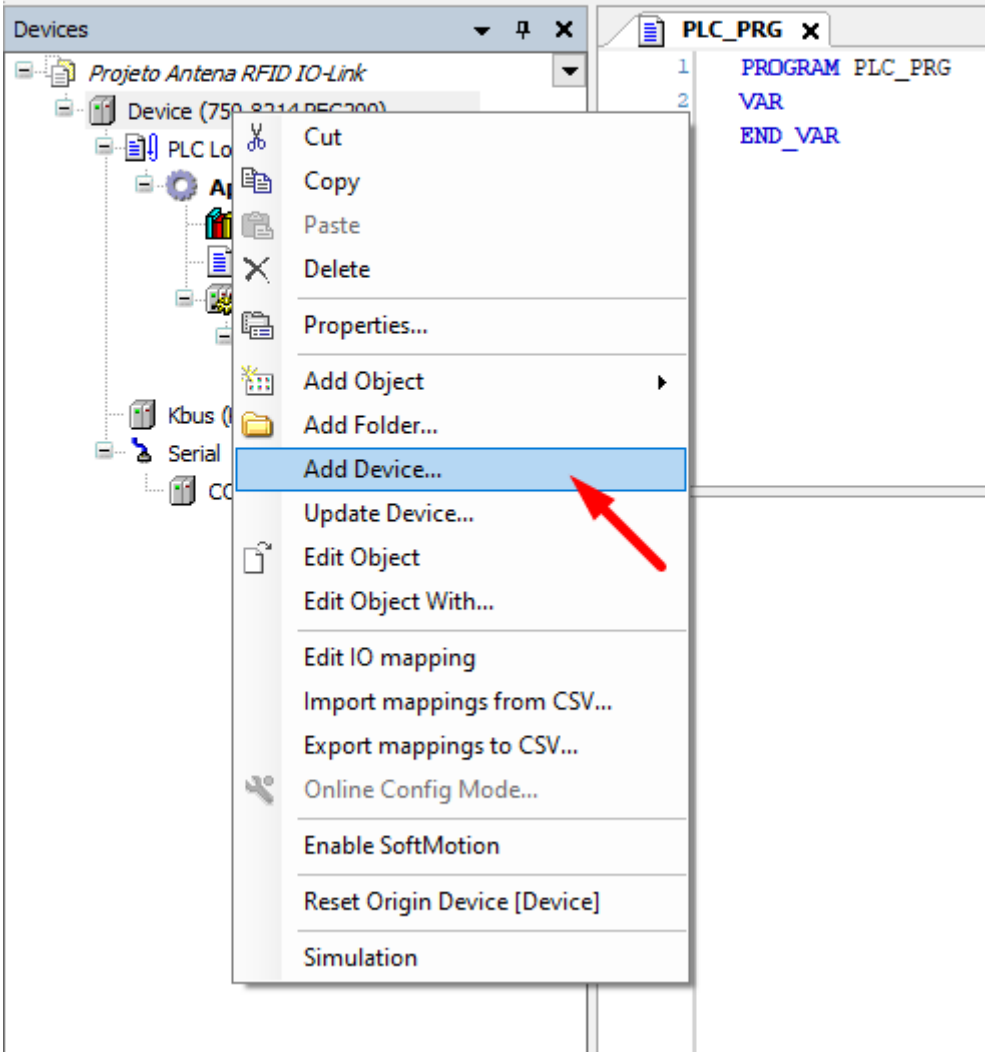
Agora, devemos adicionar a Remota IO-Link a lista de dispositivos conectados ao nosso CLP, clique sobre ele com o botão direito, selecione a opção *Add Device* e adicione um adaptador Ethernet.

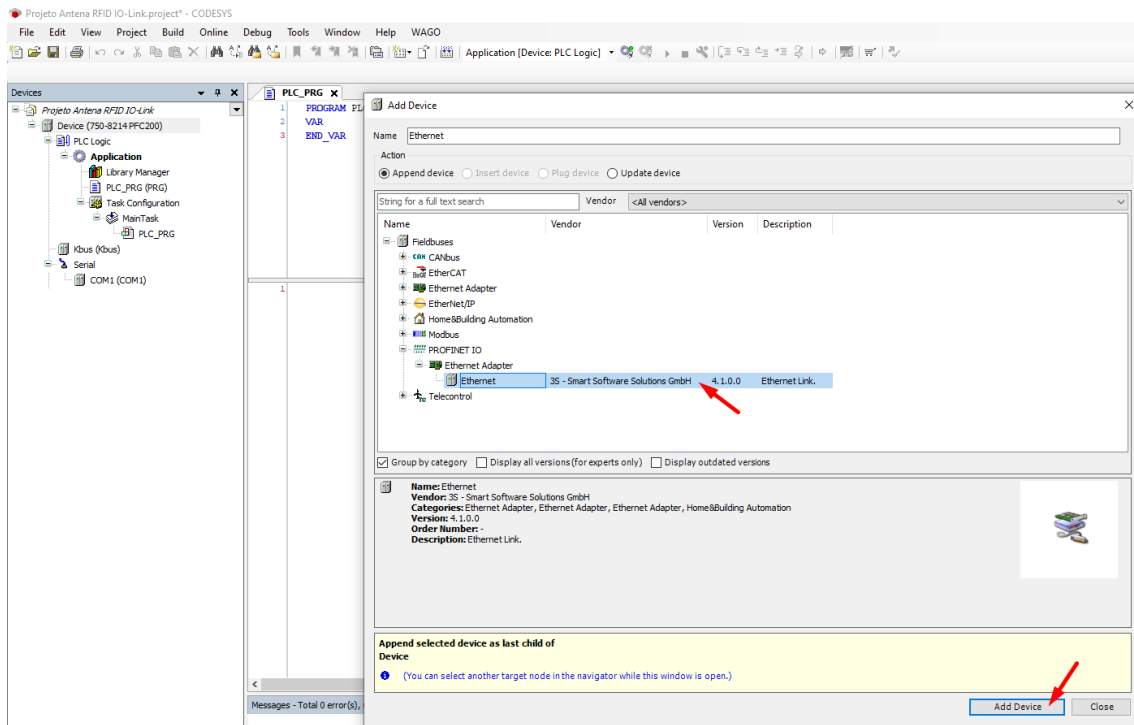
Devices

- Projeto Antena RFID IO-Link
 - Device (750-8214-05C200)
 - PLC Lo
 - Properties...
 - Add Object
 - Add Folder...
 - Add Device...**
 - Update Device...
 - Edit Object
 - Edit Object With...
 - Edit IO mapping
 - Import mappings from CSV...
 - Export mappings to CSV...
 - Online Config Mode...
 - Enable SoftMotion
 - Reset Origin Device [Device]
 - Simulation

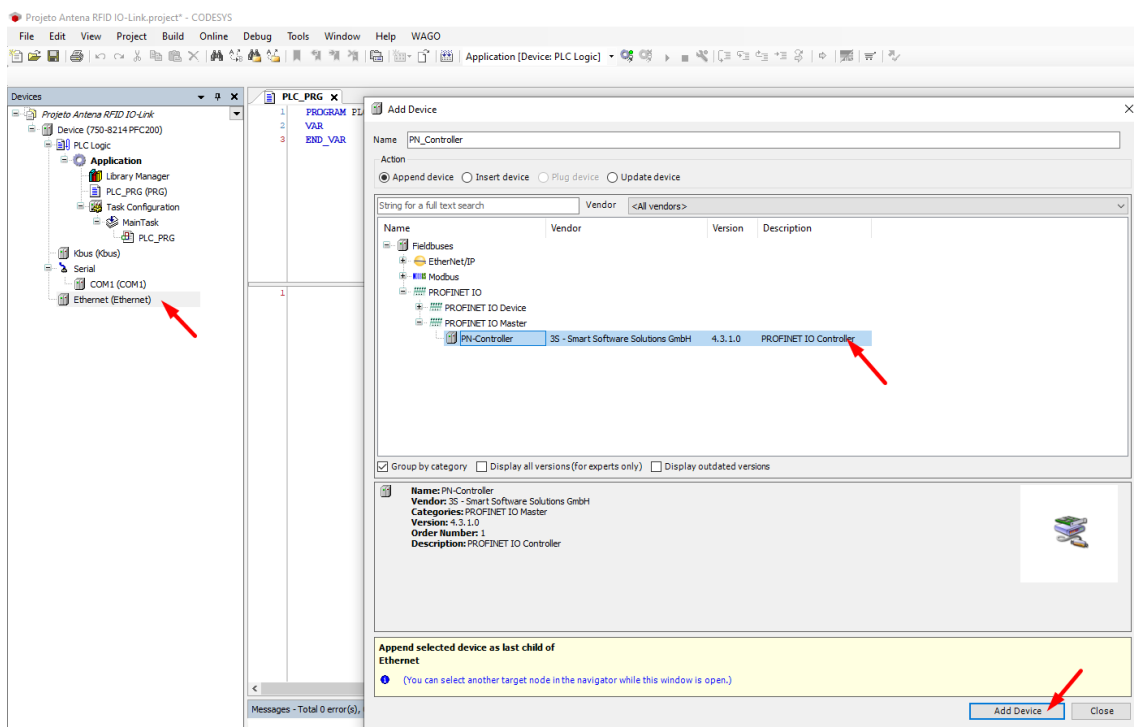
PLC_PRG x

```
1 PROGRAM PLC_PRG
2 VAR
END_VAR
```

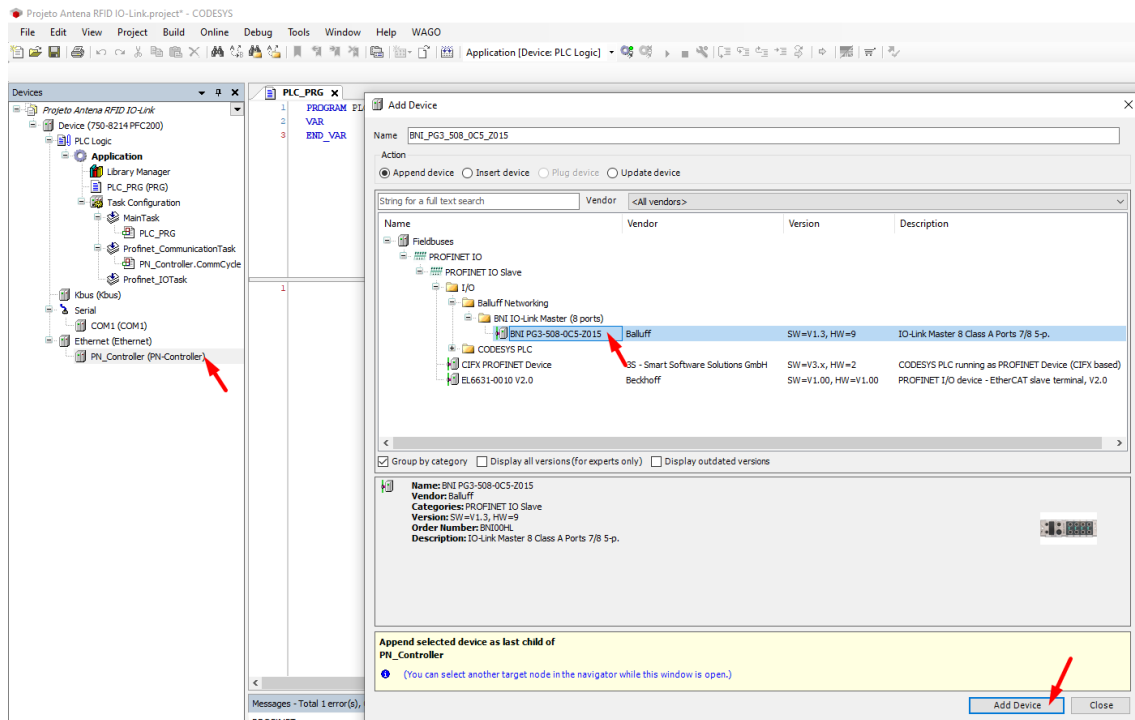




Em seguida, dê um duplo clique sobre o adaptador Ethernet na lista de dispositivos e adicione um controlador Profinet.

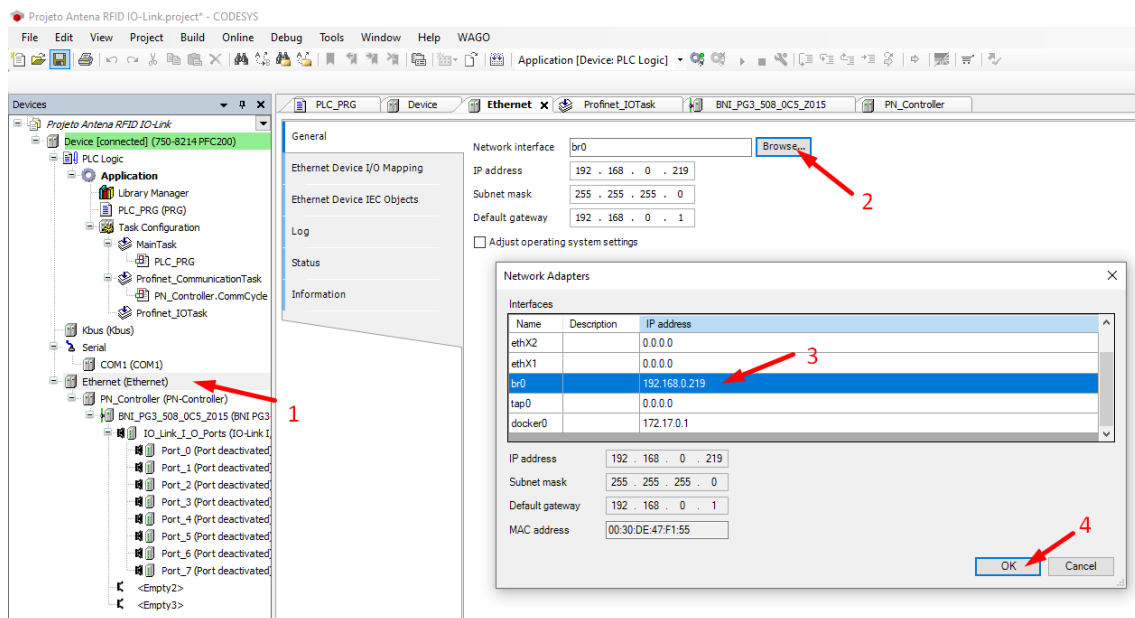


Posteriormente, selecione com um clique duplo o controlador Profinet na lista de dispositivos, expanda a lista *PROFINET IO Slave > I/O > Balluff Networking > BNI IO-Link Master* e adicione a remota *BNI PG3-508-OC5-Z015* ao projeto.

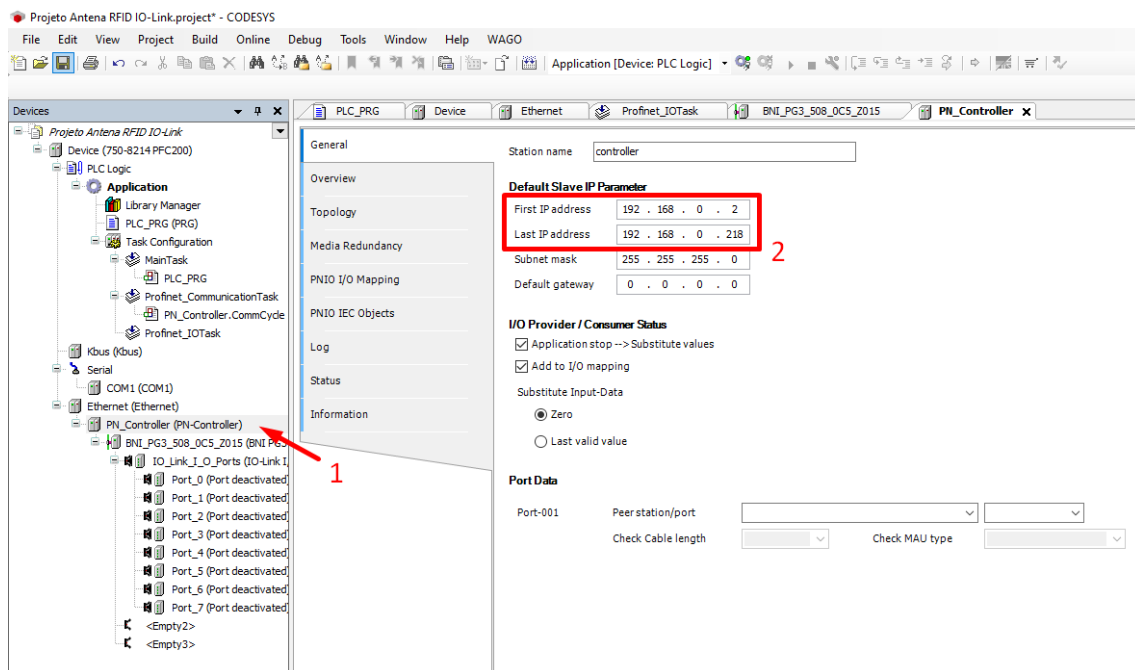


3 – Configurando a interface de rede da remota IO-Link

Com o CLP conectado ao seu computador, acesse o adaptador Ethernet e aponte a interface de rede para o IP do CLP.

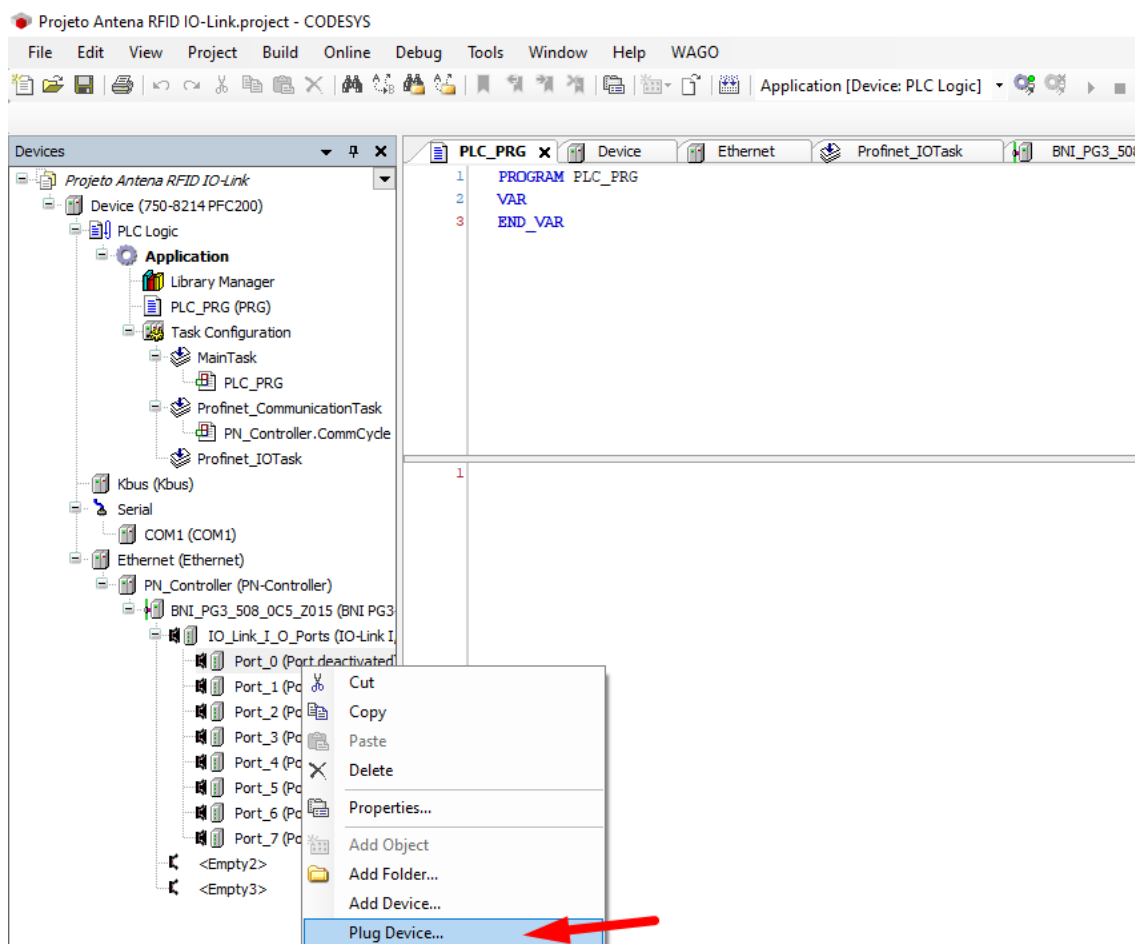


Na sequência, selecione o controlador Profinet com um clique duplo e ajuste a faixa de endereçamento IP dos dispositivos escravos.

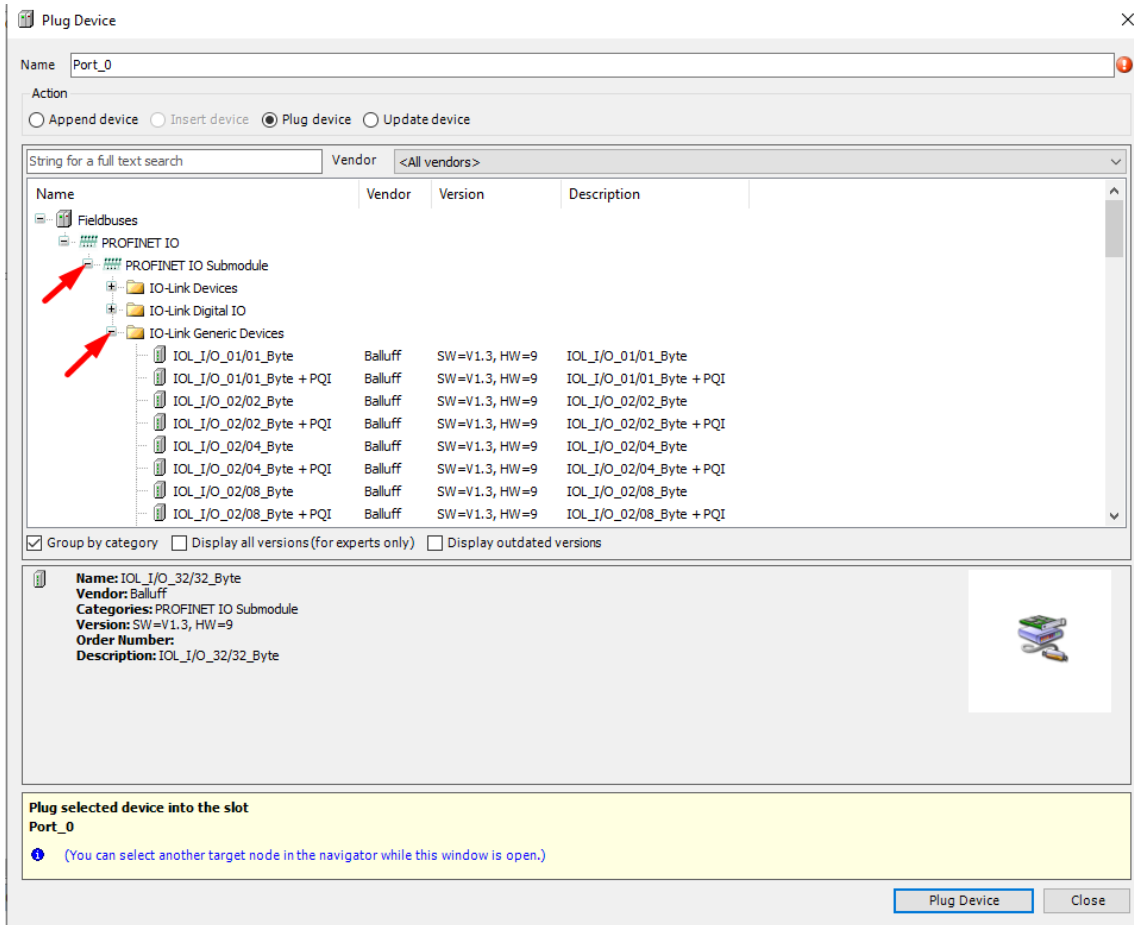


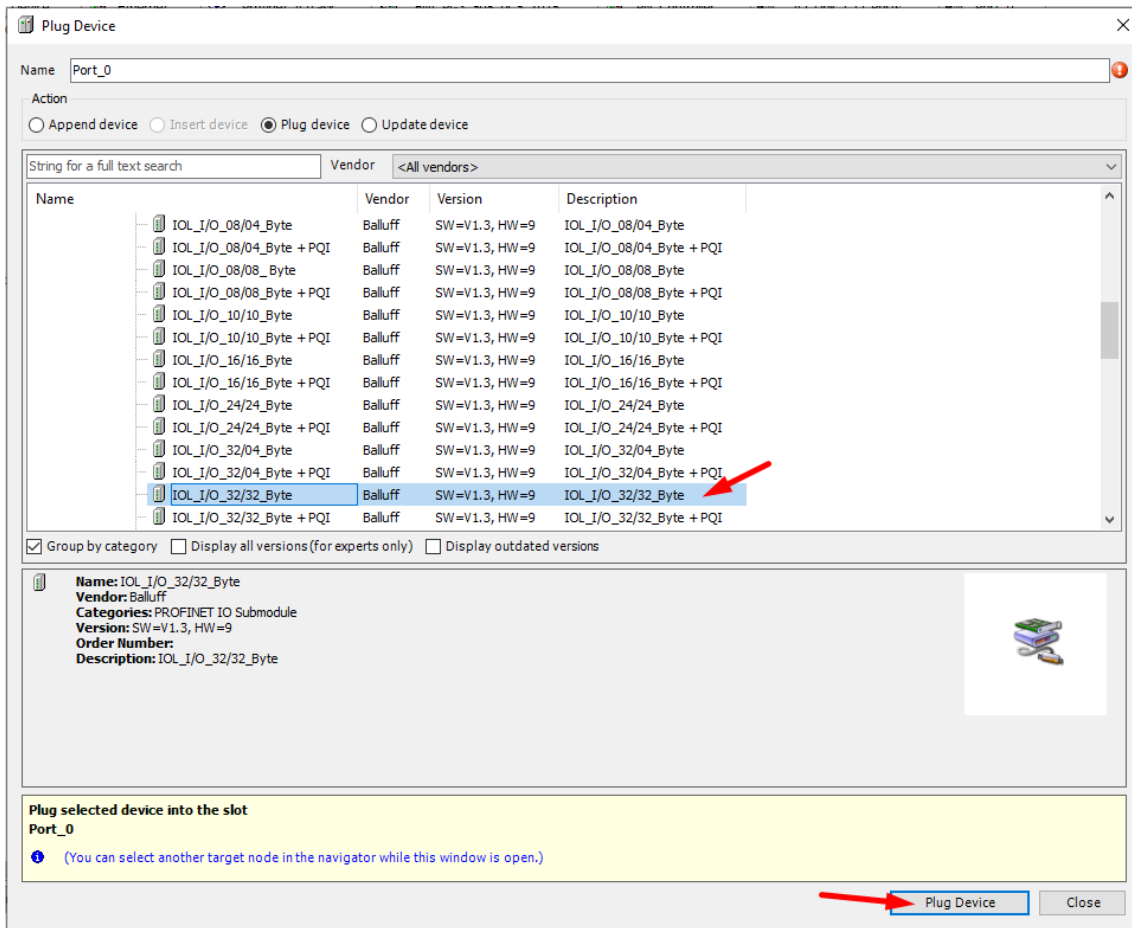
4 – Adicionando a antena IO-Link ao projeto

Devemos especificar em qual porta a antena IO-Link será conectada. Assim, clique com o botão direito sobre a porta desejada e selecione a opção *Plug Device*.



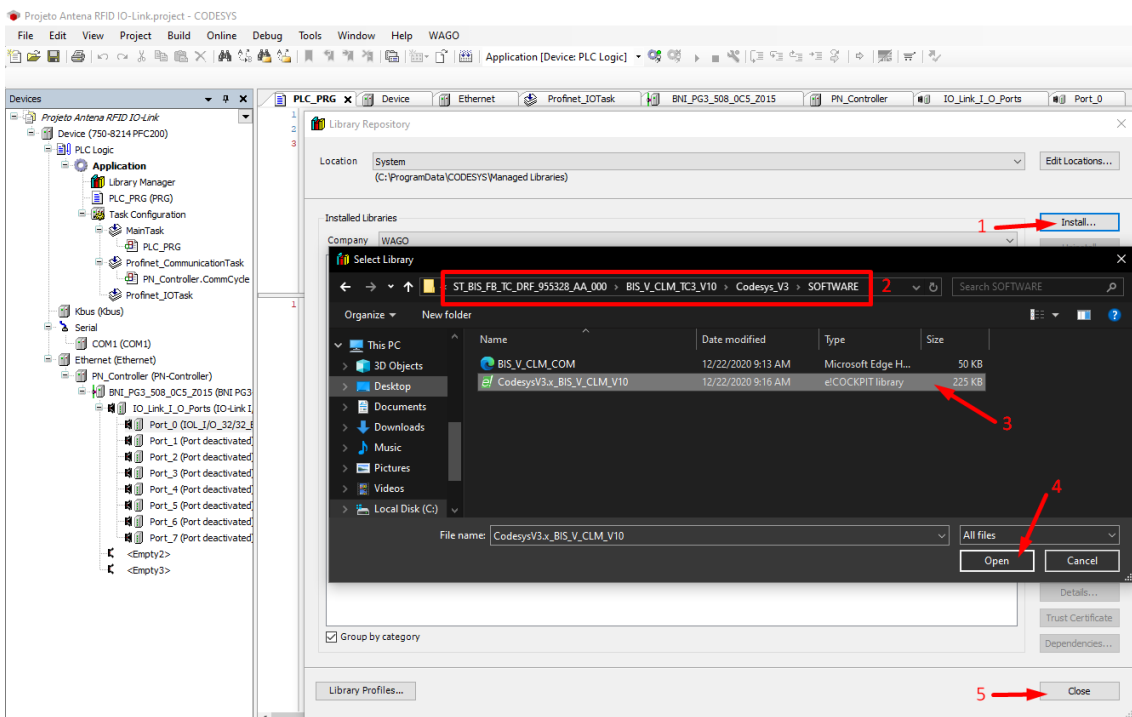
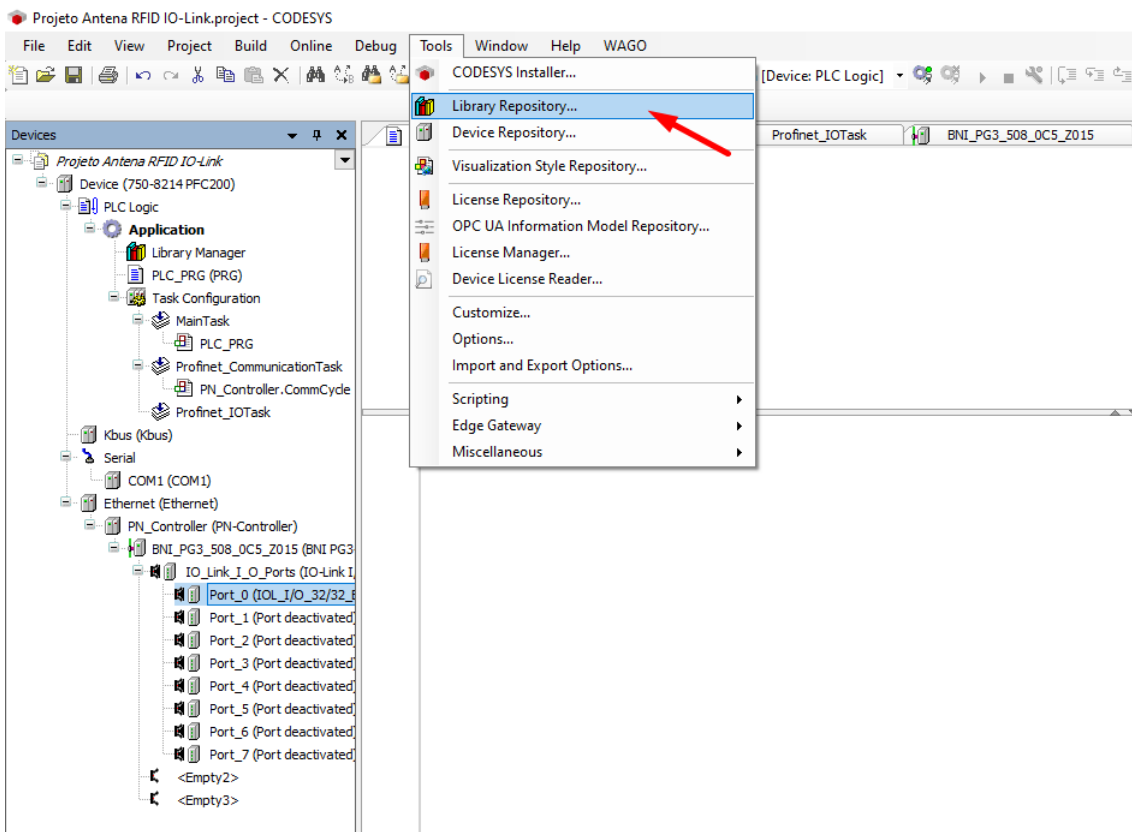
Uma nova janela será aberta, expanda as listas *PROFINET IO Submodule* e *IO-Link Generic Devices*, escolha o dispositivo *IOL_I/O_32/32_Byte* e clique em *Plug Device*.



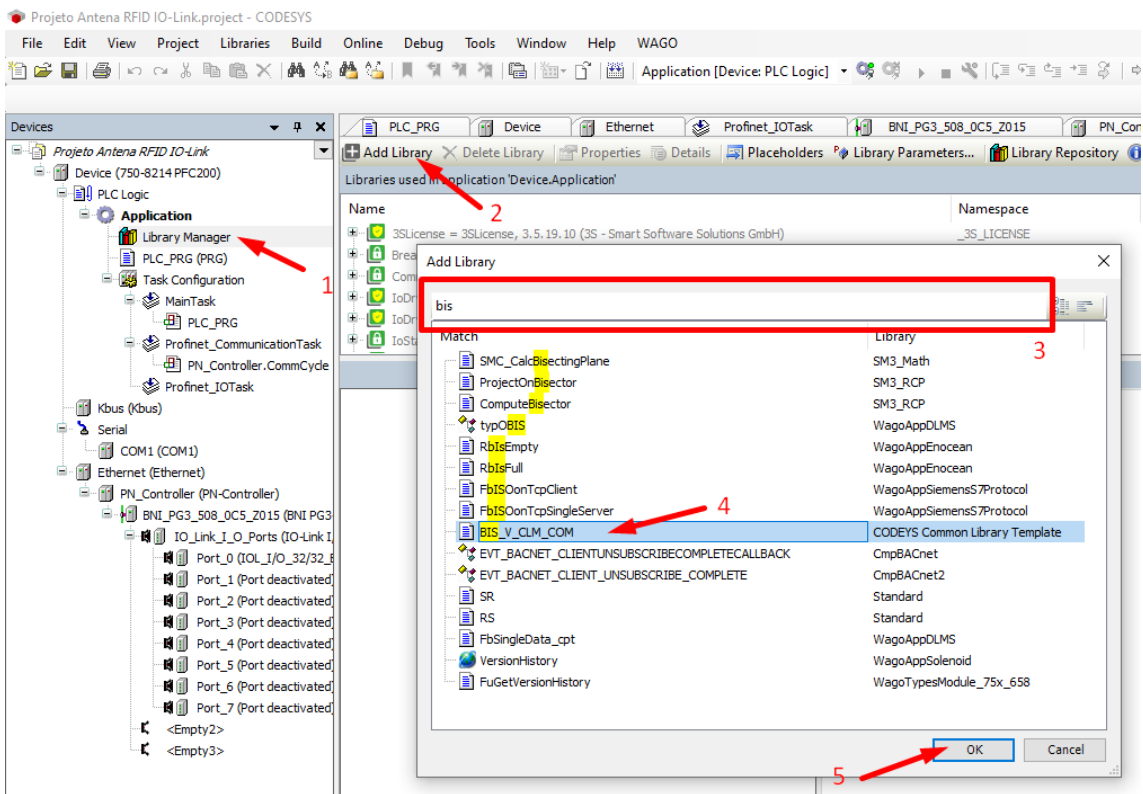


5 – Adicionando o function block da antena

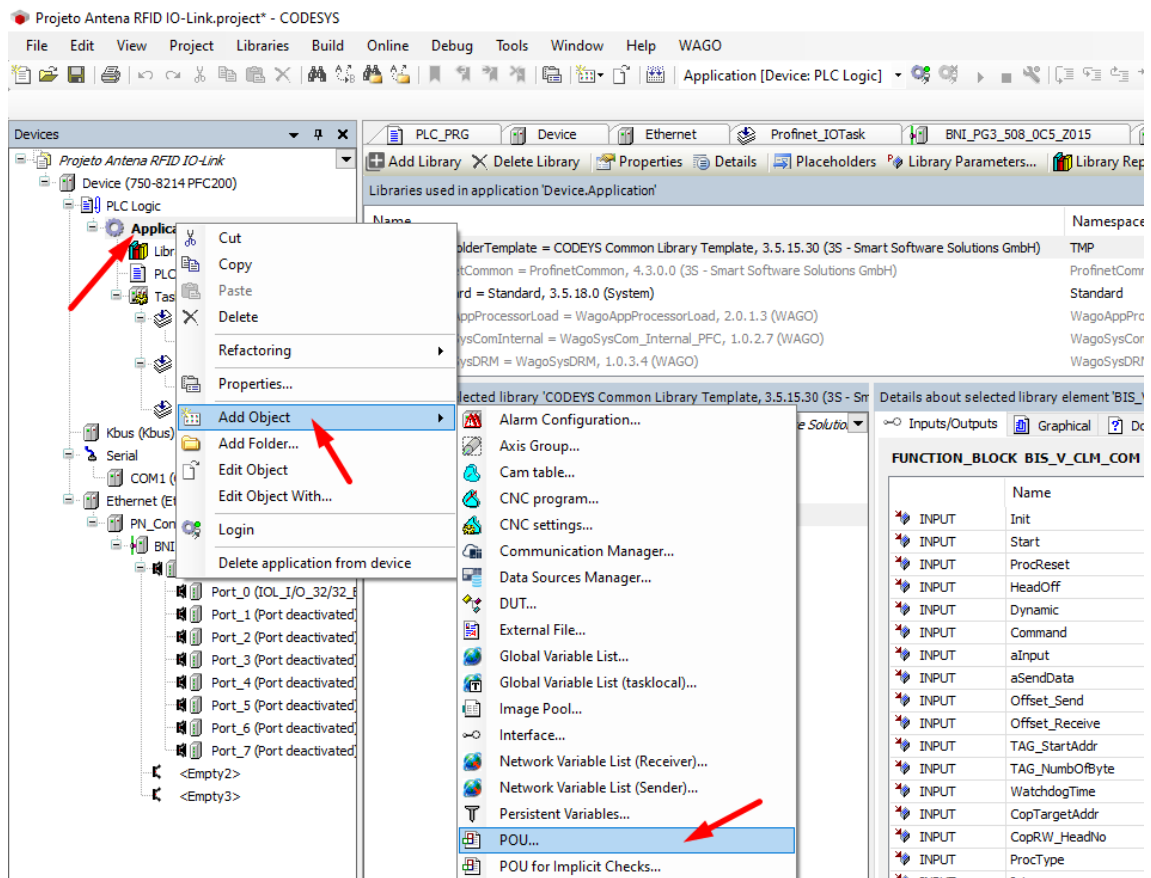
Agora, devemos adicionar o componente que é responsável por gerenciar as operações de leitura e escrita da antena nas bibliotecas do Codesys. Desse modo, acesse o repositório de bibliotecas no menu Tools e clique em *Install*. Na sequência, entre na pasta contendo os arquivos da antena baixados anteriormente, selecione o item nomeado *CodesysV3.x_BIS_V_CLM_V10* e finalize clicando em *Open*.



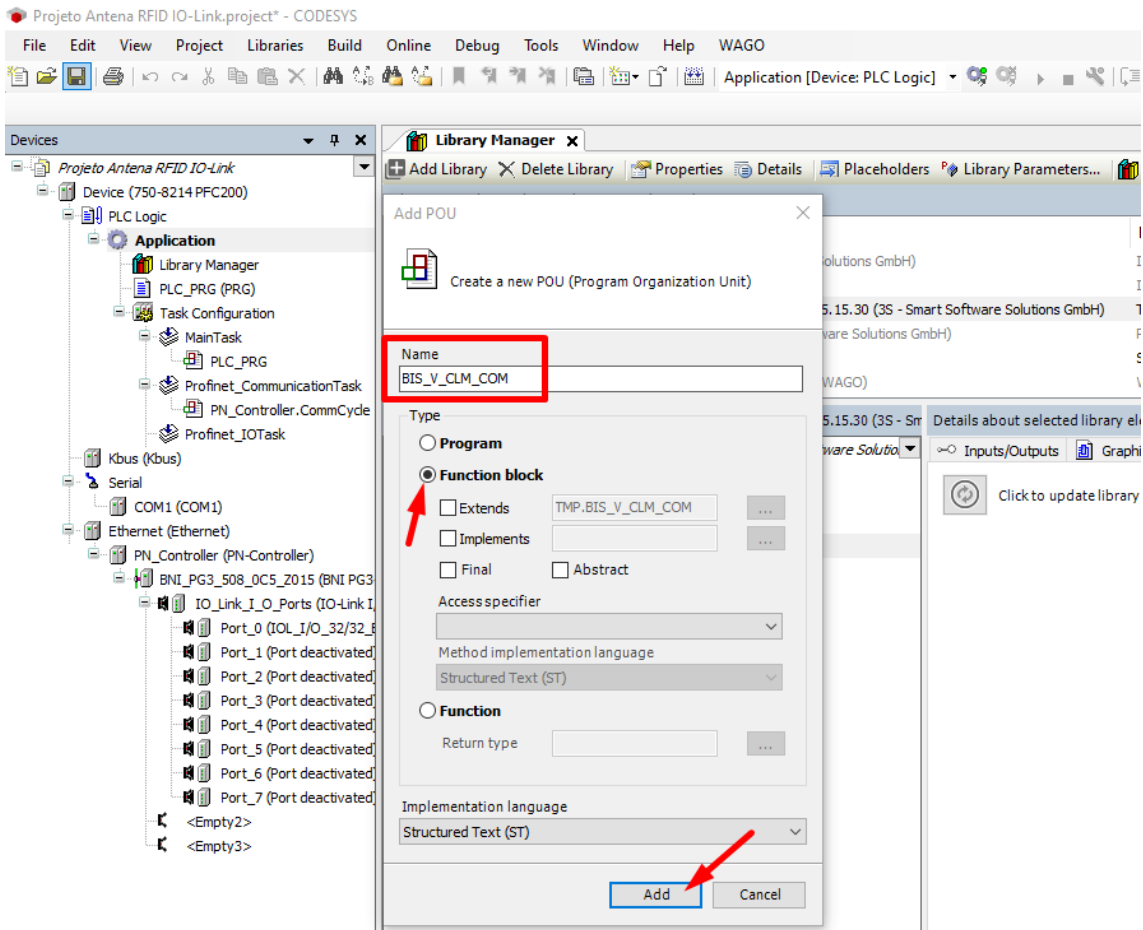
Em seguida, devemos adicionar a biblioteca ao projeto atual, entre no *Library Manager* e clique na opção *Add Library*. Na nova janela que se abriu, preencha o campo de pesquisa com a palavra “bis”, selecione o bloco *BIS_V_CLM_COM* e finalize no botão *OK*.



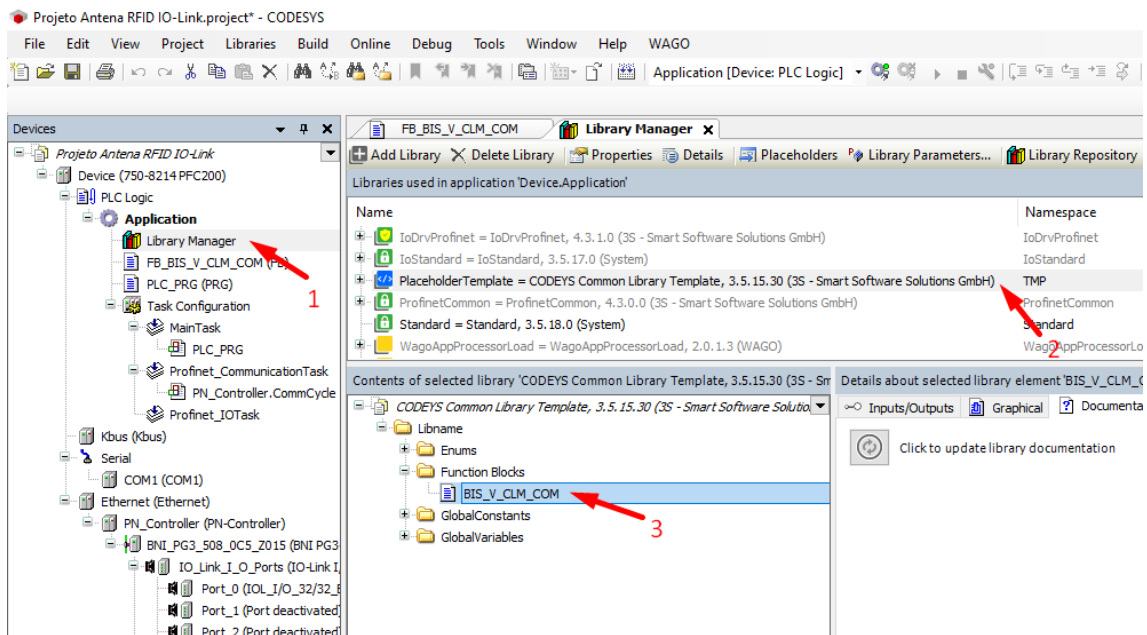
Nesse momento, já estamos com o bloco de função adicionado ao nosso projeto, todavia, esse bloco foi compilado para uma versão antiga do Codesys, tornando-o incompatível com as bibliotecas atuais do programa. A fim de evitarmos conflitos, devemos criar um novo bloco de função idêntico ao bloco da biblioteca. Para isso, clique com o botão direito na *Application* na lista de dispositivos e selecione *Add Object > POU*.

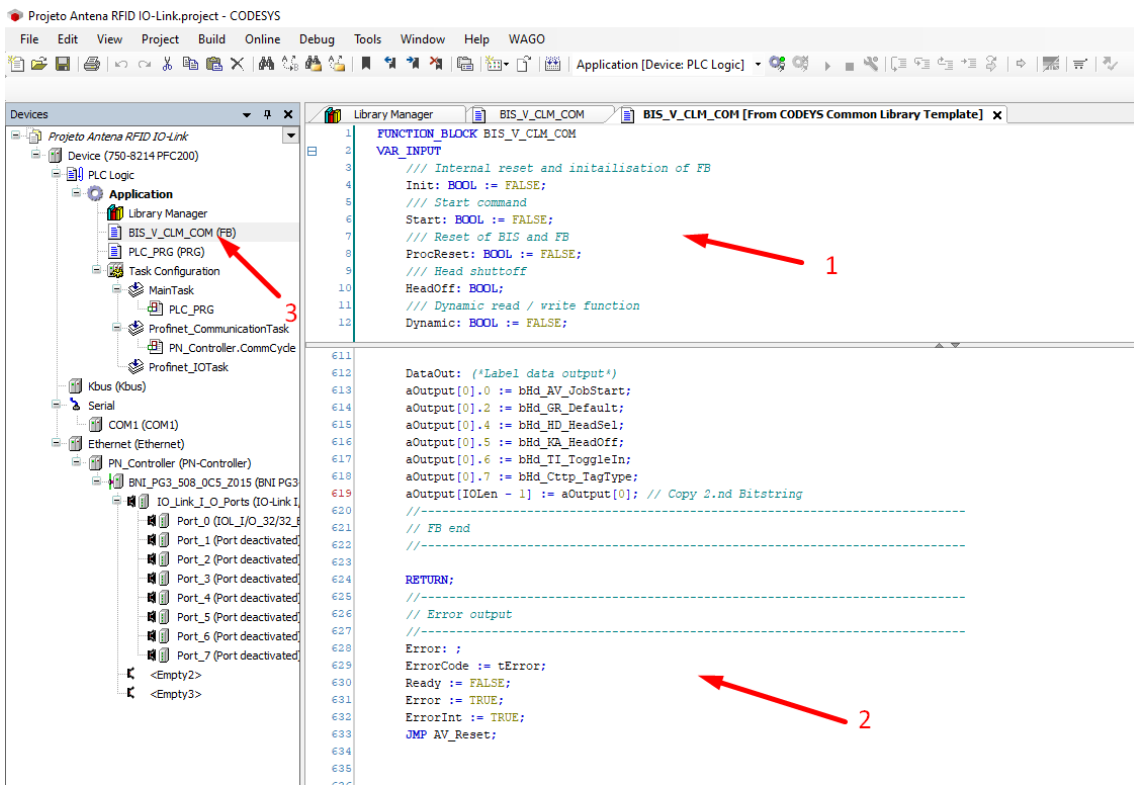


Com a nova janela aberta, preencha o nome como “BIS_V_CLM_COM”, marque a opção *Function Block* e finalize em *Add*.

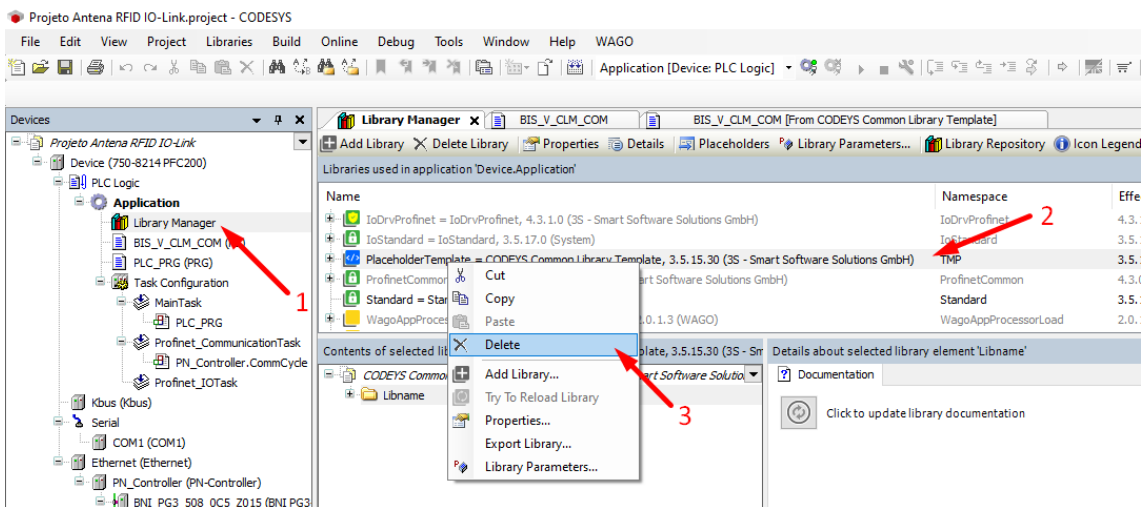


Agora, devemos copiar na totalidade o bloco de função da biblioteca para o bloco que acabamos de criar. Assim, acesse o *Library Manager* na lista de dispositivos, procure a biblioteca CODEYS Common Library Template versão 3.5.15.30, abra o function block *BIS_V_CLM_COM* e copie tudo para o bloco criado no passo anterior.



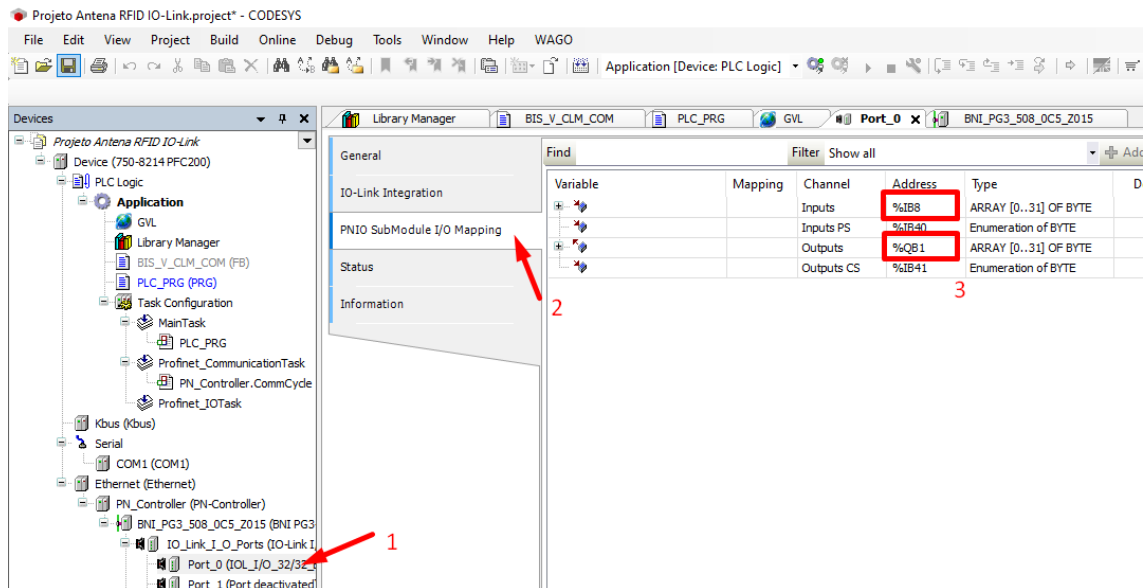


Terminado esse processo, podemos voltar ao *Library Manager* e remover a biblioteca que adicionamos anteriormente.

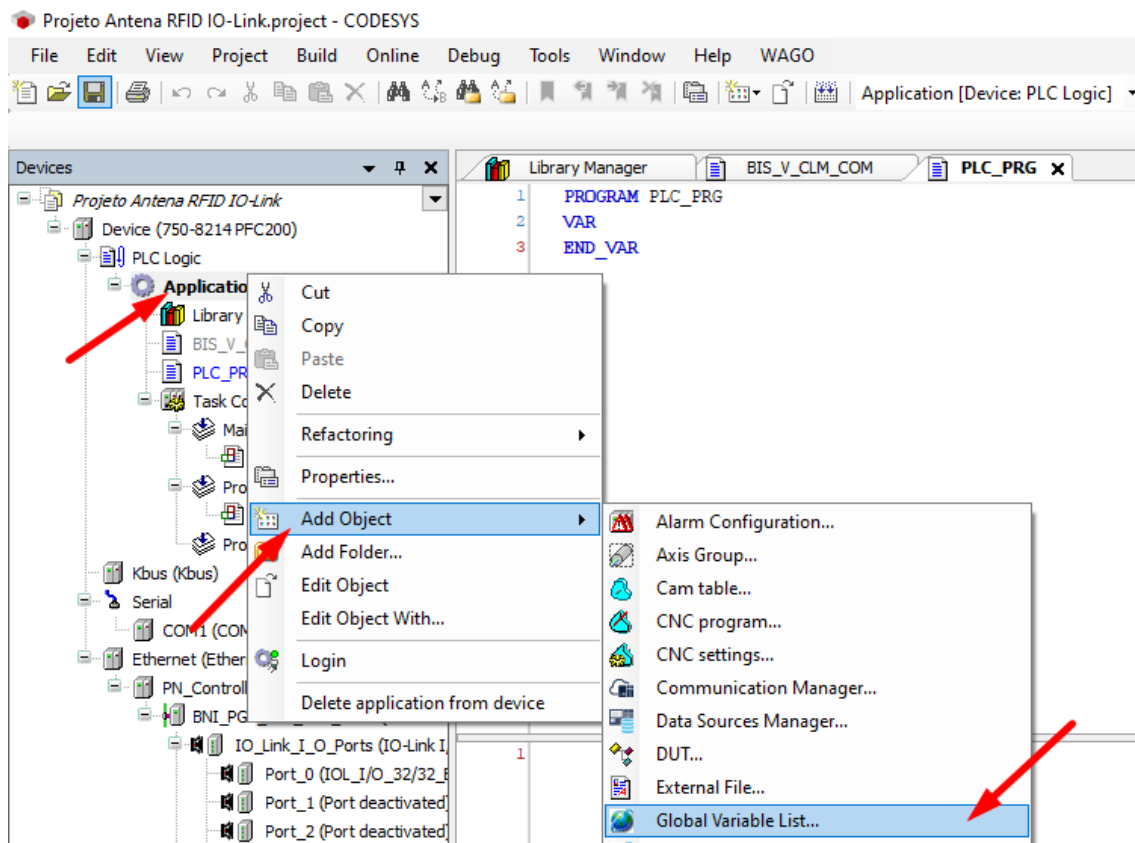


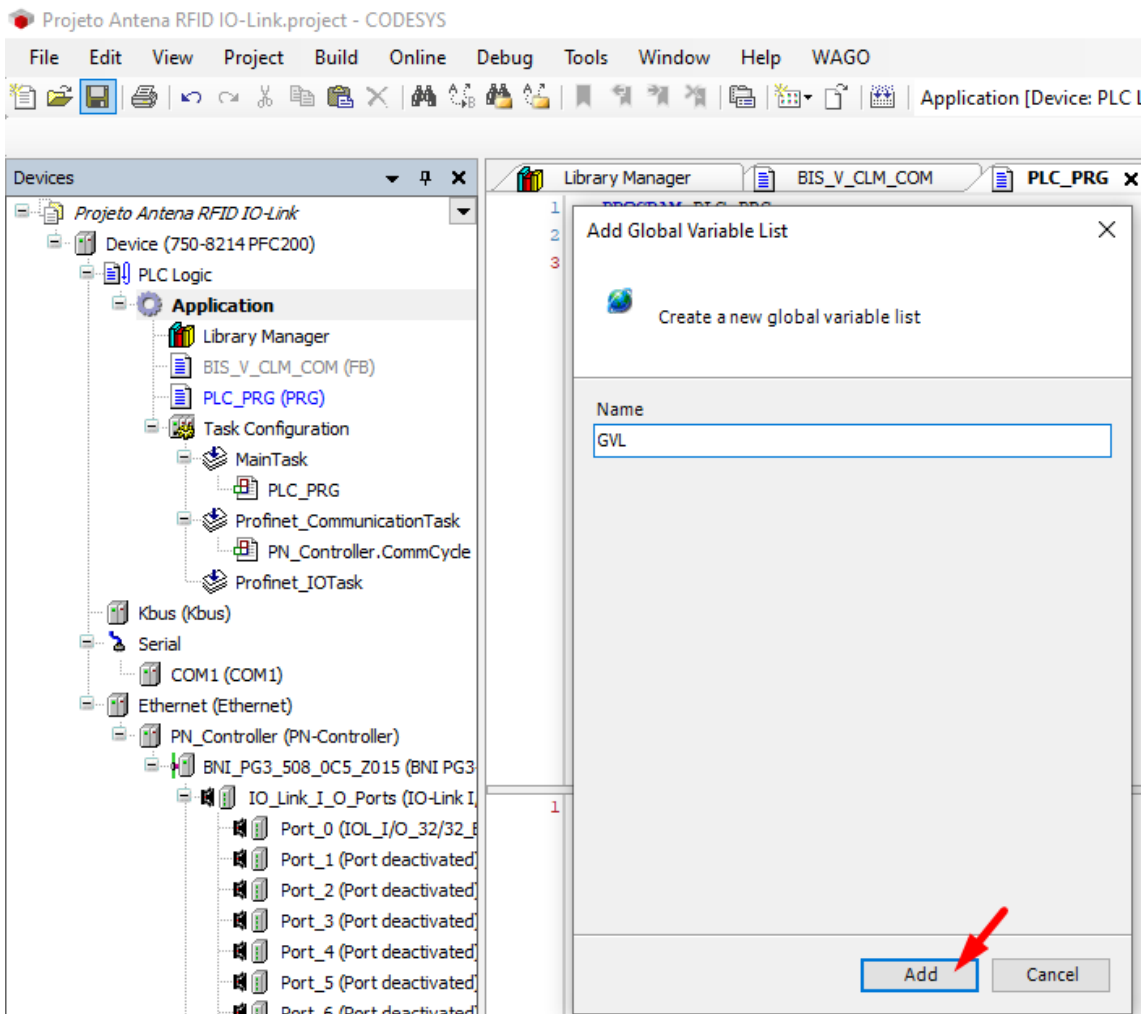
6 – Configurando as variáveis do function block da antena

Por fim, devemos configurar as variáveis que serão usadas pelo function block para garantir a comunicação e operação correta da antena RFID. Primeiramente, selecione a porta que antena está conectada na lista de *Devices*, entre no menu *PNIO SubModule I/O Mapping* e anote os endereços iniciais dos buffers de entrada e saída da antena.

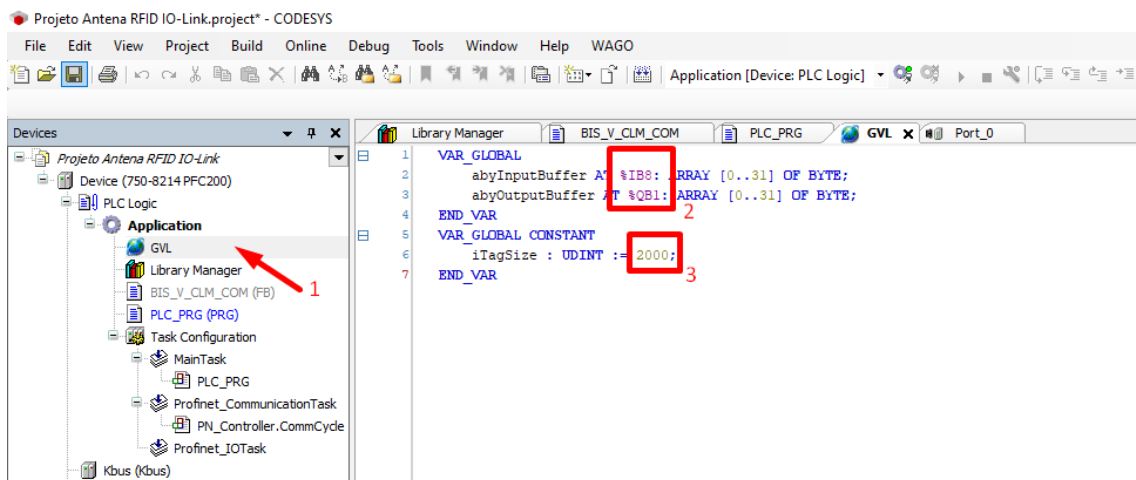


Em seguida, adicione uma lista de variáveis globais clicando com o botão direito sobre *Application* e selecione as opções *Add Object > Global Variable List*. Na janela que abrir, apenas finalize clicando em *Add*.

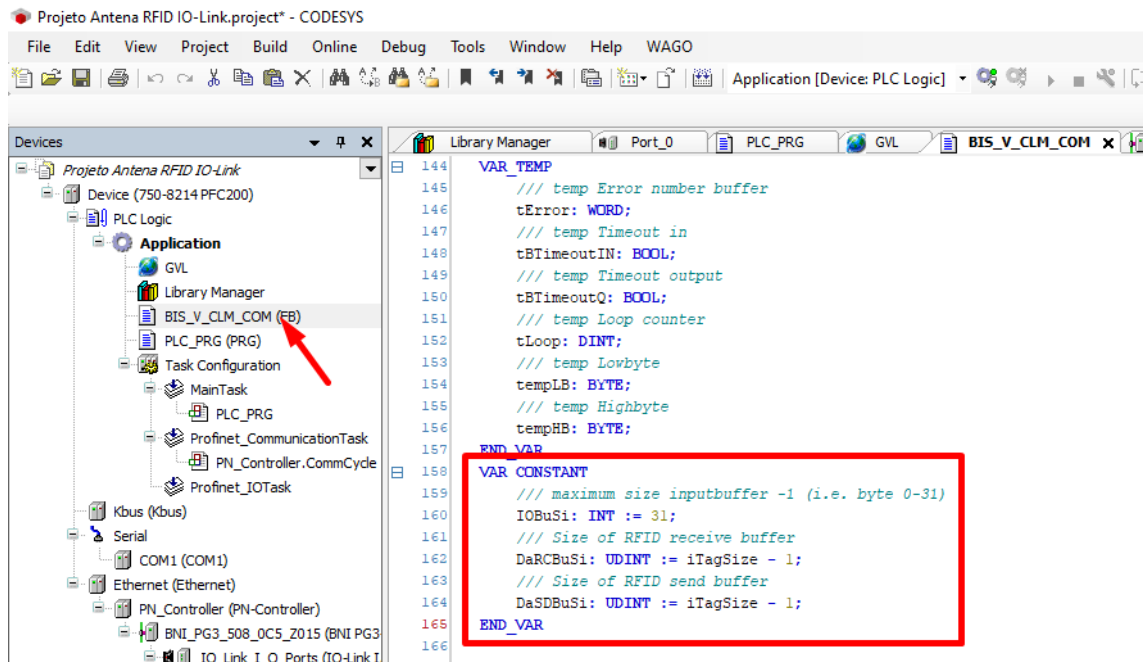




Agora, abra a lista que acabamos de criar e declare as variáveis correspondentes aos bytes de entrada e saída da antena, garantindo que elas apontem para os endereços de memória verificados no início desta seção. Além disso, é preciso incluir uma constante com o tamanho da tag utilizada na aplicação, como no nosso exemplo está sendo usada a tag BIS0045, atribuímos o valor de 2000 bytes. Desse modo, o código ficará da seguinte forma:



Finalizada a criação das variáveis globais, temos que alterar parte do código do function block *BIS_V_CLM_COM*. Acesse o bloco e altere as variáveis *IOBuSi*, *DaRCBuSi* e *DaSDBuSi* para “31”, “iTagSize - 1” e “iTagSize - 1”, respectivamente. Ficando da seguinte forma:



Na sequência, vamos instanciar o bloco de função da antena e criar todas as variáveis necessárias para o seu correto funcionamento. Assim, acesse o programa principal e cole a seguinte declaração de variáveis:

```

PROGRAM PLC_PRG
VAR
    bis_v_clm : BIS_V_CLM_COM;
    R_Ready: R_TRIG;

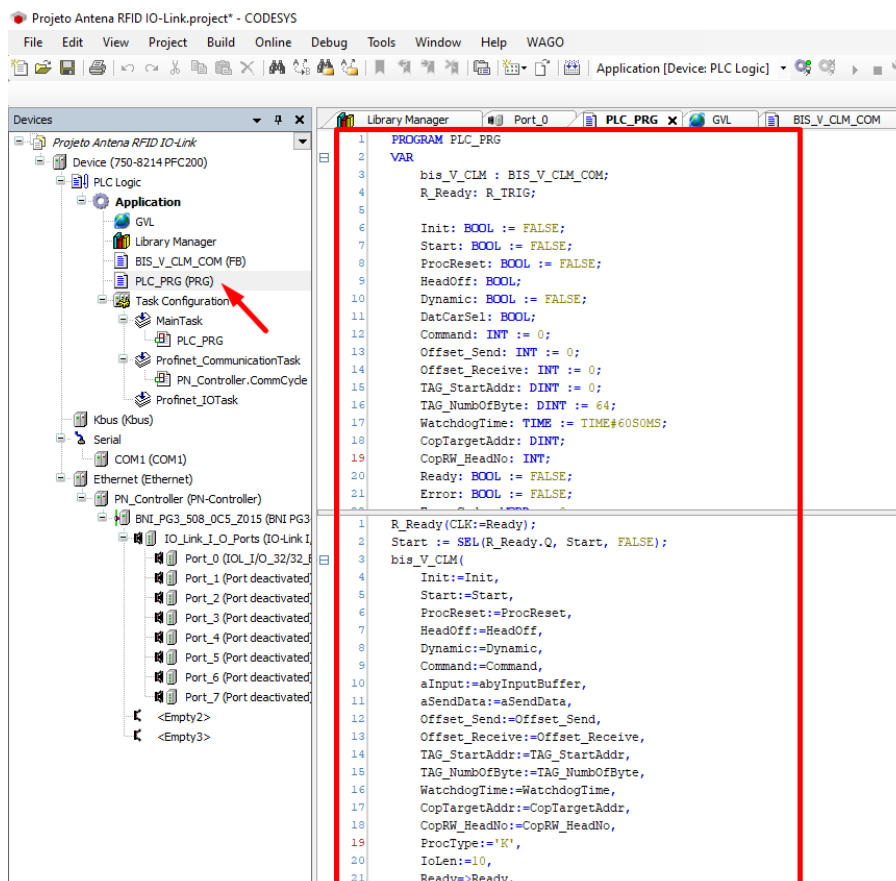
    Init: BOOL := FALSE;
    Start: BOOL := FALSE;
    ProcReset: BOOL := FALSE;
    HeadOff: BOOL;
    Dynamic: BOOL := FALSE;
    DatCarSel: BOOL;
    Command: INT := 0;
    Offset_Send: INT := 0;
    Offset_Receive: INT := 0;
    TAG_StartAddr: DINT := 0;
    TAG_NumbOfByte: DINT := 64;
    WatchdogTime: TIME := TIME#60S0MS;
    CopTargetAddr: DINT;
    CopRW_HeadNo: INT;
    Ready: BOOL := FALSE;
    Error: BOOL := FALSE;
    ErrorCode: WORD := 0;
    MT: BOOL := FALSE;
    DatCarrPresent: BOOL := FALSE;
    aSendData: ARRAY[0..iTagSize-1] OF BYTE;
    aReceiveData: ARRAY[0..iTagSize-1] OF BYTE;
END_VAR

```

Do mesmo modo, cole o seguinte código na seção de programa:

```
R_Ready(CLK:=Ready);
Start := SEL(R_Ready.Q, Start, FALSE);
bis_V_CLM(
  Init:=Init,
  Start:=Start,
  ProcReset:=ProcReset,
  HeadOff:=HeadOff,
  Dynamic:=Dynamic,
  Command:=Command,
  aInput:=abyInputBuffer,
  aSendData:=aSendData,
  Offset_Send:=Offset_Send,
  Offset_Receive:=Offset_Receive,
  TAG_StartAddr:=TAG_StartAddr,
  TAG_NumbOfByte:=TAG_NumbOfByte,
  WatchdogTime:=WatchdogTime,
  CopTargetAddr:=CopTargetAddr,
  CopRW_HeadNo:=CopRW_HeadNo,
  ProcType:='K',
  IoLen:=10,
  Ready=>Ready,
  Error=>Error,
  ErrorCode=>ErrorCode,
  MT=>MT,
  DatCarrPresent=>DatCarrPresent,
  aOutput=>abyOutputBuffer,
  aReceiveData=>aReceiveData);
```

Com essas adições, seu programa principal deve ficar da seguinte forma:



```
1 PROGRAM PLC_PRG
2
3 VAR
4   bis_V_CLM : BIS_V_CLM_COM;
5   R_Ready: R_TRIG;
6
7   Init: BOOL := FALSE;
8   Start: BOOL := FALSE;
9   ProcReset: BOOL := FALSE;
10  HeadOff: BOOL;
11  Dynamic: BOOL := FALSE;
12  DatCarSel: BOOL;
13  Command: INT := 0;
14  Offset_Send: INT := 0;
15  Offset_Receive: INT := 0;
16  TAG_StartAddr: DINT := 0;
17  TAG_NumbOfByte: DINT := 64;
18  WatchdogTime: TIME := TIME#60S0MS;
19  CopTargetAddr: DINT;
20  CopRW_HeadNo: INT;
21  Ready: BOOL := FALSE;
22  Error: BOOL := FALSE;
23
24 R_Ready(CLK:=Ready);
25 Start := SEL(R_Ready.Q, Start, FALSE);
26 bis_V_CLM(
27   Init:=Init,
28   Start:=Start,
29   ProcReset:=ProcReset,
30   HeadOff:=HeadOff,
31   Dynamic:=Dynamic,
32   Command:=Command,
33   aInput:=abyInputBuffer,
34   aSendData:=aSendData,
35   Offset_Send:=Offset_Send,
36   Offset_Receive:=Offset_Receive,
37   TAG_StartAddr:=TAG_StartAddr,
38   TAG_NumbOfByte:=TAG_NumbOfByte,
39   WatchdogTime:=WatchdogTime,
40   CopTargetAddr:=CopTargetAddr,
41   CopRW_HeadNo:=CopRW_HeadNo,
42   ProcType:='K',
43   IoLen:=10,
44   Ready=>Ready,
45   Error=>Error,
46   ErrorCode=>ErrorCode,
47   MT=>MT,
48   DatCarrPresent=>DatCarrPresent,
49   aOutput:=abyOutputBuffer,
50   aReceiveData=>aReceiveData);
```

7 – Como Realizar Escrita e Leitura Através do Bloco

Com todos itens do projeto configurados, chegou a hora de testar na prática a comunicação entre a antena e uma tag RFID.

Primeiramente, faça o login no CLP através do comando *Alt+F8* e pressione *F5* para iniciar o programa. Em seguida, devemos manipular as seguintes variáveis:

- *Start* – comando para iniciar a operação da antena
- *Command* – determina qual operação será realizada no sistema
- *TAG_StartAddr* – endereço de memória em que a operação será executada
- *TAG_NumbOfByte* – número de bytes que serão utilizados na operação
- *aSendData* – array que contém os dados a serem gravados na tag
- *aReceiveData* – array que recebe os dados lidos da tag

Ou seja, para realizar uma operação de escrita de 32 bytes no endereço de memória 0. Devemos atribuir os seguintes valores as variáveis:

- *Start* – TRUE
- *Command* – 2
- *TAG_StartAddr* – 0
- *TAG_NumbOfByte* – 32
- *aSendData* – dados que deseja gravar

A seguir, apresentamos um exemplo que escreve 255 nos primeiros 4 bytes de memória de uma tag e 0 nos bytes restantes:

The screenshot shows the CODESYS environment for a project named 'Projeto Antena RFID IO-Link'. The 'Variable Declaration' table is visible, listing various variables and their values. The 'Start' variable is highlighted with a red box and set to 'TRUE'. Other variables like 'Command' are set to '2', and 'aSendData' is an array with values 255, 0, 0, 0, 0.

Expression	Type	Value	Prepared value
* bis_V_CLM	BIS_V_CLM_COM		
* R_Ready	R_TRIG		
Init	BOOL	FALSE	
Start	BOOL	FALSE	TRUE
ProcReset	BOOL	FALSE	
HeadOff	BOOL	FALSE	
Dynamic	BOOL	FALSE	
DatCarSel	BOOL	FALSE	
Command	INT	0	2
Offset_Send	INT	0	
Offset_Receive	INT	0	
TAG_StartAddr	DINT	0	0
TAG_NumbOfByte	DINT	0	32
WatchdogTime	TIME	T#1m	
CopTargetAddr	DINT	0	
CopRW_HeadNo	INT	0	
Ready	BOOL	TRUE	
Error	BOOL	FALSE	
ErrorCode	WORD	0	
MT	BOOL	FALSE	
DatCarrPresent	BOOL	FALSE	
aSendData	ARRAY [0..(TagSize...]		
aSendData[0]	BYTE	0	255
aSendData[1]	BYTE	0	255
aSendData[2]	BYTE	0	255
aSendData[3]	BYTE	0	255
aSendData[4]	BYTE	0	255

Para a leitura o processo é bem similar, com a diferença de que o valor do comando deve ser alterado para 1 e os dados da tag são transferidos para o array *aReceiveData*.

Chegamos ao fim! Agora, é só seguir os passos acima e implementar seu sistema RFID conforme as necessidades da sua aplicação.