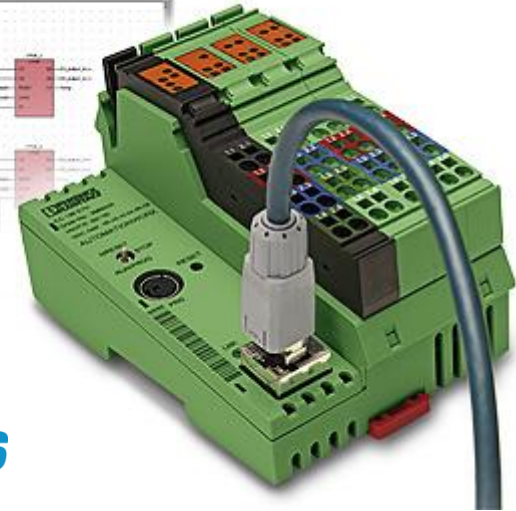
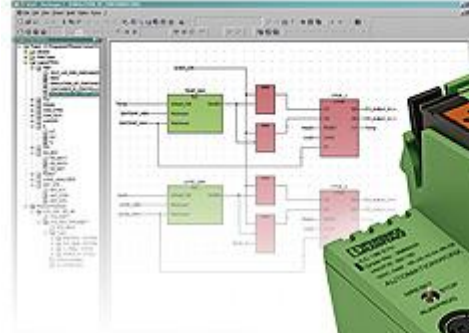
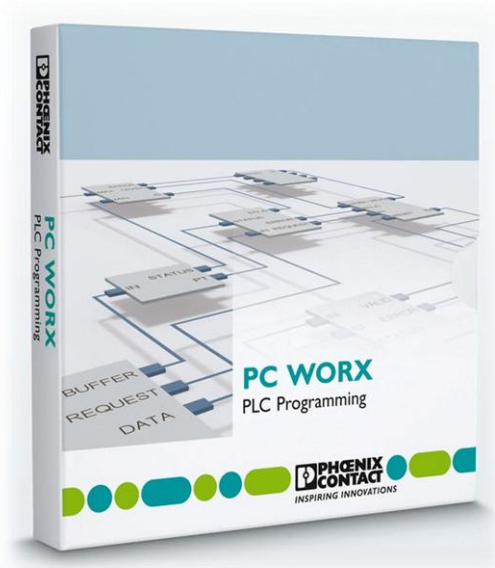


# CONFIGURAÇÃO DE COMUNICAÇÃO MODBUS TCP/IP ENTRE DOIS CONTROLADORES



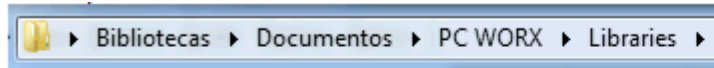
**Via Biblioteca Modbus**

## • DICAS – Trabalhando com bibliotecas da PHOENIX CONTACT

### 1. O que devo fazer quando copiar uma biblioteca no formato zwt(compactada)?

Deve salvar o arquivo dentro da pasta "Libraries" conforme caminho indicado abaixo e após isso abrir esta no **PCWORX**(**não no PCWORX Express**) e apenas compilar.

Dependendo da versão de windows o locais deve ser estes:



### 2. O que devo fazer quando baixar um biblioteca do site da Phoenix Contact?

Ao baixar do site a biblioteca, ela virá em arquivo de instalação, sendo assim basta apontar para que a mesma já seja instalada no local certo. Após isso basta abrir e compilar.

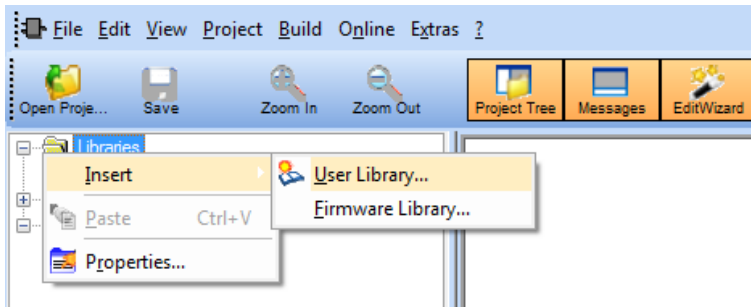
## • DICAS – Trabalhando com bibliotecas da PHOENIX CONTACT

### 3. O que devo fazer quando a biblioteca chama outra biblioteca?

Faça o mesmos passos anteriormente indicados porém abrindo e compilando primeiro aquela biblioteca que é chamada dentro da biblioteca que você pretende usar.

### 4. Como importar uma biblioteca para o meu projeto?

Basta clicar com o botão direito sobre a pasta Libraries e buscar a biblioteca pretendida.



## • DICAS – Trabalhando com bibliotecas da PHOENIX CONTACT

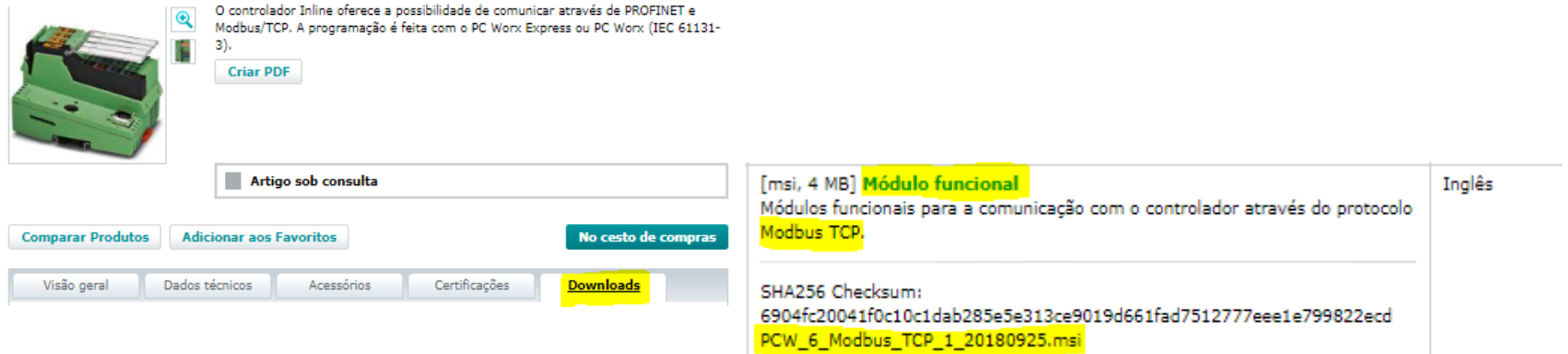
### 5. Importou a biblioteca, mas ao compilar esta dando erro relacionado a mesma?

Se você importou a biblioteca e a mesma apresenta um \*(asterisco) ao lado do nome, significa que a mesma não foi compilada corretamente.

### 6. Como encontrar e baixar no site da Phoenix Contact as bibliotecas para PCWORX?

Entre no campo de busca com o código de qualquer controlador da Phoenix Contact e vá na aba download.

Controle - ILC 131 ETH - 2700973



O controlador Inline oferece a possibilidade de comunicar através de PROFINET e Modbus/TCP. A programação é feita com o PC Worx Express ou PC Worx (IEC 61131-3).

[Criar PDF](#)

Artigo sob consulta

[Comparar Produtos](#) [Adicionar aos Favoritos](#) [No cesto de compras](#)

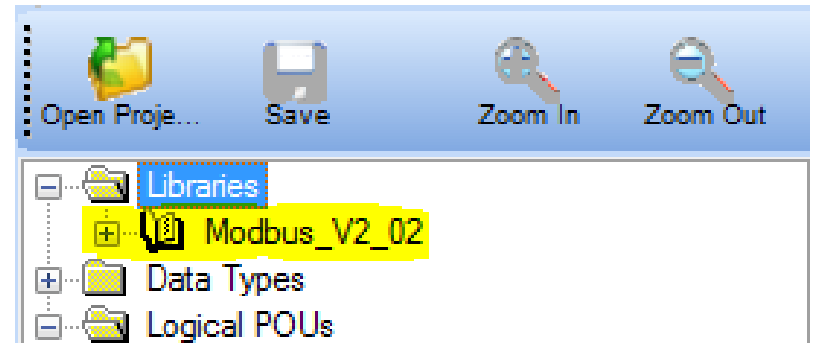
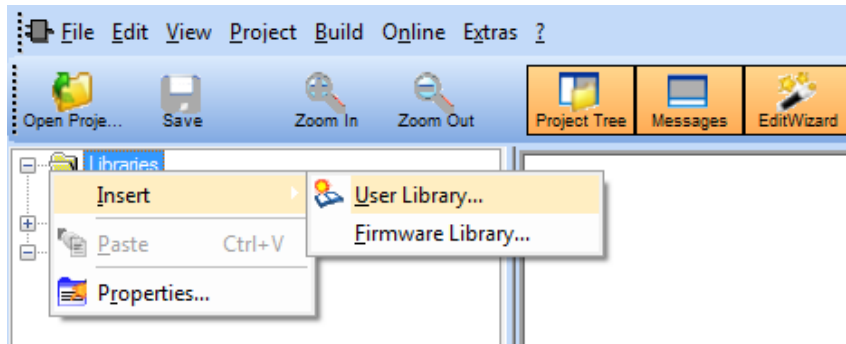
[Visão geral](#) [Dados técnicos](#) [Acessórios](#) [Certificações](#) [Downloads](#)

[msi, 4 MB] **Módulo funcional**  
Módulos funcionais para a comunicação com o controlador através do protocolo Modbus TCP.

Inglês

SHA256 Checksum:  
6904fc20041f0c10c1dab285e5e313ce9019d661fad7512777eee1e799822ecd  
**PCW\_6\_Modbus\_TCP\_1\_20180925.msi**

- IEC Programming Workspace – Adicionar a biblioteca Modbus ao Projeto.



Atualmente, já possui disponível no site para download gratuitamente versões mais atualizadas.

# • IEC Programming Workspace – Adicionar os blocos de comunicação

## Configuração do Client(mestre) Modbus TCP/IP

The screenshot displays the IEC Programming Workspace interface. On the left, the 'Libraries' pane shows the project structure with 'Modbus\_V2\_02' and 'Modbus\_TCP\_IP' folders. The main workspace contains three Modbus TCP/IP client blocks:

- MB\_TCP\_Client\_V2\_0\_3** (top center):
  - Inputs: TRUE (xActivate), TRUE (xAutoAck), '192.168.05' (strServer\_IP), 50 (Port), FALSE (xUDP\_Mode), t#5s (tTimeout), udt\_MB\_Com\_Data (udtMB\_Com\_Data).
  - Outputs: xActive (MB\_Ativo), xError (wDiagCode), wDiagCode (wAddDiagCode).
- MB\_TCP\_FC16\_V2\_00\_1** (bottom left):
  - Inputs: MB\_Ativo (xActivate), ack (xAcknowledge), 2 (IMT\_ID), t#1s (tUpdateTime), uiUnitIdentifier (uiUnitIdentifier), WORD#0 (wStartRegister), UINT#50 (uiQuantityOfRegisters), arr\_Escrita\_MB (arrRegisterValue), udt\_MB\_Com\_Data (udtTCP\_ComData).
  - Outputs: xActive (xActive), xDone (xDone), xError (wDiagCode), wDiagCode (wAddDiagCode).
- MB\_TCP\_FC3\_V2\_00\_1** (bottom right):
  - Inputs: MB\_Ativo (xActivate), ack (xAcknowledge), 1 (IMT\_ID), t#1s (tUpdateTime), uiUnitIdentifier (uiUnitIdentifier), WORD#51 (wStartRegister), UINT#50 (uiQuantityOfRegisters), udt\_MB\_Com\_Data (udtTCP\_ComData).
  - Outputs: xActive (xActive), xNDR (wDiagCode), wDiagCode (wAddDiagCode), arr\_Leitura\_MB (arrRegisterValue).

On the right, the 'Edit Wizard' pane shows the 'Name' list for the 'Modbus\_V2\_02' group. The following components are highlighted with red boxes:

- MB\_TCP\_Client\_V2\_00
- MB\_TCP\_FC16\_V2\_00
- MB\_TCP\_FC3\_V2\_00

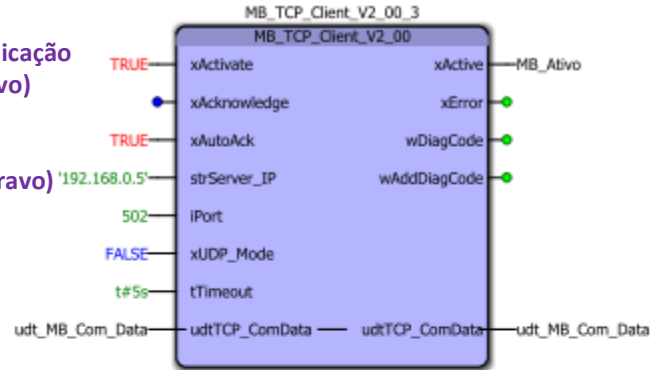
Red arrows point from these highlighted components to their respective block configurations in the main workspace.

# • IEC Programming Workspace – Configurar os blocos de comunicação

## Configuração do Client(mestre) Modbus TCP/IP

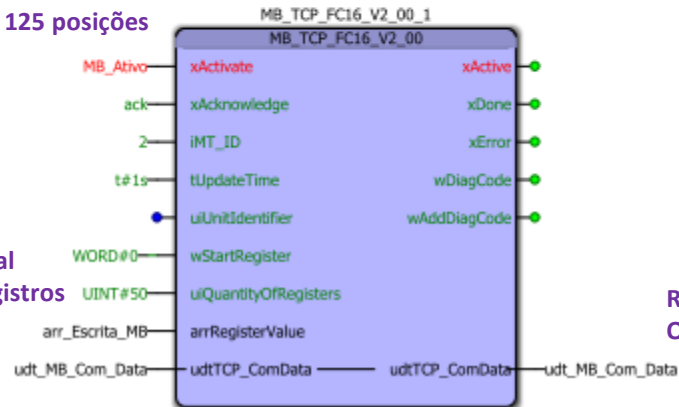
Bloco que estabelece a comunicação Modbus como “server” (escravo)

IP do server(escravo) '192.168.0.5'  
Porta



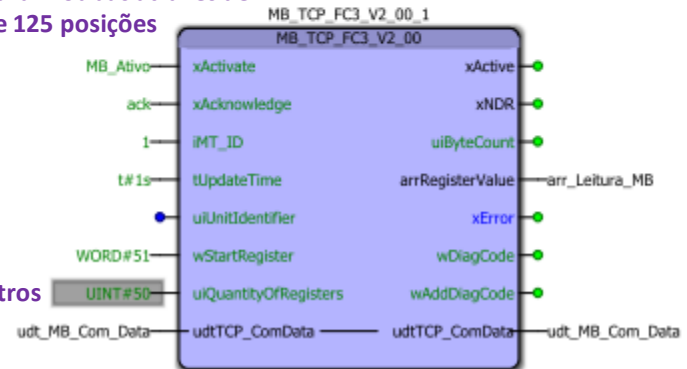
Bloco de escrita Modbus através de uma Array de 125 posições

Registro Inicial  
Quant. de registros



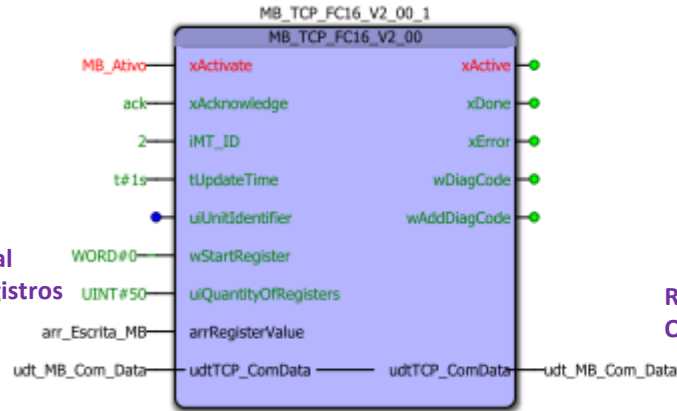
Bloco de leitura Modbus através de uma Array de 125 posições

Registro Inicial  
Quant. de registros

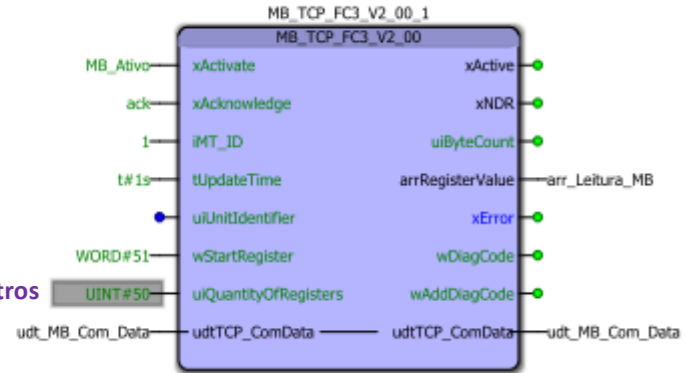


# • IEC Programming Workspace – Programa exemplo de leitura e escrita usando as arrays da comunicação Modbus TCP/IP.

Registro Inicial  
Quant. de registros



Registro Inicial  
Quant. de registros



Escrita nas posições 1 a 50 do server

Variavel\_1 ————— arr\_Escrita\_MB[1]  
 Variavel\_2 ————— arr\_Escrita\_MB[2]  
 Variavel\_3 ————— arr\_Escrita\_MB[3]  
 Variavel\_50 ————— arr\_Escrita\_MB[50]

Leitura das posições 51 a 100 do server

arr\_Leitura\_MB[1] ————— aux\_1  
 arr\_Leitura\_MB[2] ————— aux\_2  
 arr\_Leitura\_MB[2] ————— aux\_3  
 arr\_Leitura\_MB[3] ————— aux\_5



# • IEC Programming Workspace – Adicionar o bloco de comunicação

## Configuração do Server(esravo) Modbus TCP/IP

The screenshot displays the IEC Programming Workspace interface. On the left, a project tree shows the structure: Libraries > Modbus\_V2\_02 > Data Types > Logical POUs > Comunica > FB\_Ton\_Toff > Main. The main workspace shows a ladder logic network with a single block representing the Modbus TCP/IP server. The block is titled "MB\_TCP\_Server\_V2\_00\_2" and "MB\_TCP\_Server\_V2\_00".

The block's parameters are as follows:

| Parameter               | Value               | Direction |
|-------------------------|---------------------|-----------|
| xActivate               | TRUE                | Input     |
| xAcknowledge            | •                   | Input     |
| xAutoAck                | TRUE                | Input     |
| xUDP_Mode               | FALSE               | Input     |
| ITCP_Port               | 502                 | Input     |
| strPartner_IP           | '192.168.0.2'       | Input     |
| tTimeout                | t#5s                | Input     |
| uiOffsetInputRegister   | •                   | Input     |
| uiOffsetHoldingRegister | •                   | Input     |
| uiOffsetInputs          | •                   | Input     |
| uiOffsetCoils           | •                   | Input     |
| arr_MB_Data             | arrModbusData       | Input     |
| xActive                 | •                   | Output    |
| xConnected              | ONBOARD_OUTPUT_BIT0 | Output    |
| xError                  | •                   | Output    |
| wDiagCode               | •                   | Output    |
| wAddDiagCode            | •                   | Output    |
| arrModbusData           | arr_MB_Data         | Output    |

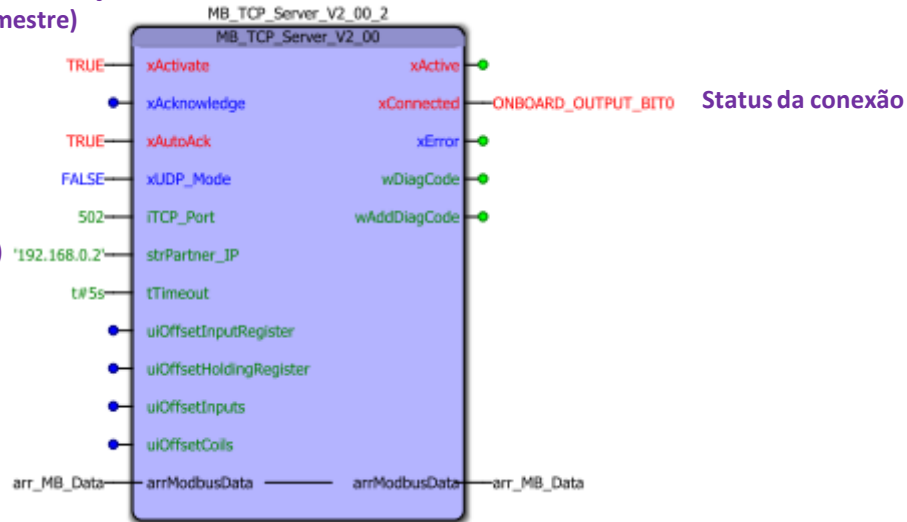
A red arrow points from the "wAddDiagCode" output of the block to the "Edit Wizard" window on the right. The "Edit Wizard" window shows a list of objects under the group "<Modbus\_V2\_02>". The object "MB\_TCP\_Server\_V2\_00" is highlighted with a red box.

# • IEC Programming Workspace – Adicionar os blocos de comunicação

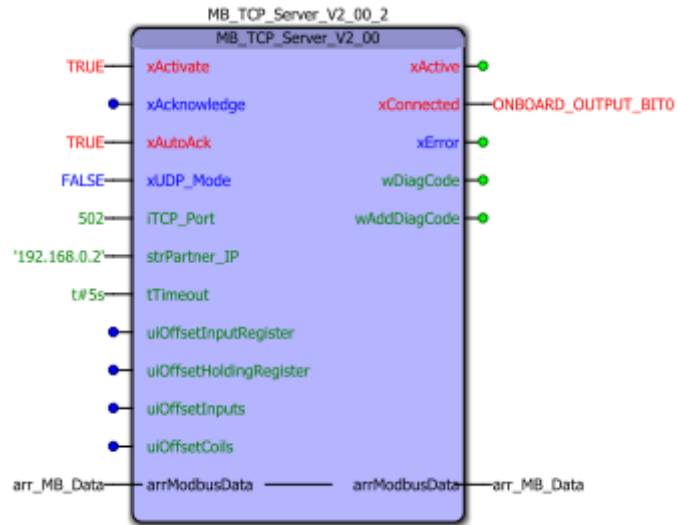
## Configuração do Server(escravo) Modbus TCP/IP

Bloco que estabelece a comunicação Modbus como “client” (mestre)

Porta  
IP do client(mestre)



# • IEC Programming Workspace – Programa exemplo de leitura e escrita usando as arrays da comunicação Modbus TCP/IP.



Array de 0 a 7167 posições  
Independente de ser leitura ou escrita, quem determina a função é o client(mestre)

## Recebendo dados do client nas posições 0 a 50

arr\_MB\_Data[0] ————— Variavel\_1  
arr\_MB\_Data[1] ————— Variavel\_2  
arr\_MB\_Data[2] ————— Variavel\_3  
arr\_MB\_Data[49] ————— Variavel\_4

## Disponibilizando dados ao client nas posições 51 a 100

ONBOARD\_INPUT\_BIT0 ————— arr\_MB\_Data[51].X0  
ONBOARD\_INPUT\_BIT1 ————— arr\_MB\_Data[52].X0  
ONBOARD\_INPUT\_BIT2 ————— arr\_MB\_Data[53].X0  
ONBOARD\_INPUT\_BIT3 ————— arr\_MB\_Data[100].X0

# Pronto!!!

A comunicação Modbus TCP/IP já está configurada.  
Agora você já pode iniciar a programação.



***INSPIRING INNOVATIONS***

[www.phoenixcontact.com.br](http://www.phoenixcontact.com.br)