

# Setting the AXC F 2152 as a Modbus Client

## Objective:

This document will provide instructions on how to download the Modbus TCP library, add that library into PLCnext Engineer, and set the PLCnext Controller (AXC F 2152) up as a Modbus TCP client. This guide will use PLCnext Engineer 2021.0, IL ETH BK DI8 DO4 2TX-PAC – 2703981, and the AXC F 2152 – 2404267.

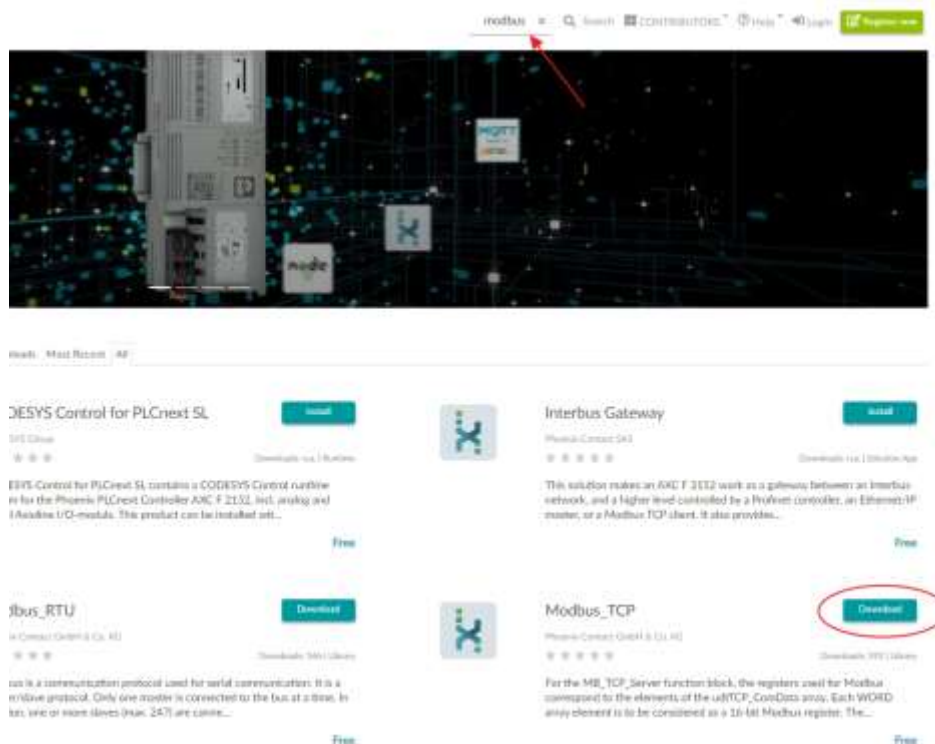
## Prerequisites:

Basic understanding of PLCnext Engineer and the Modbus TCP protocol.

## Procedure:

### A. Download the Modbus TCP library.

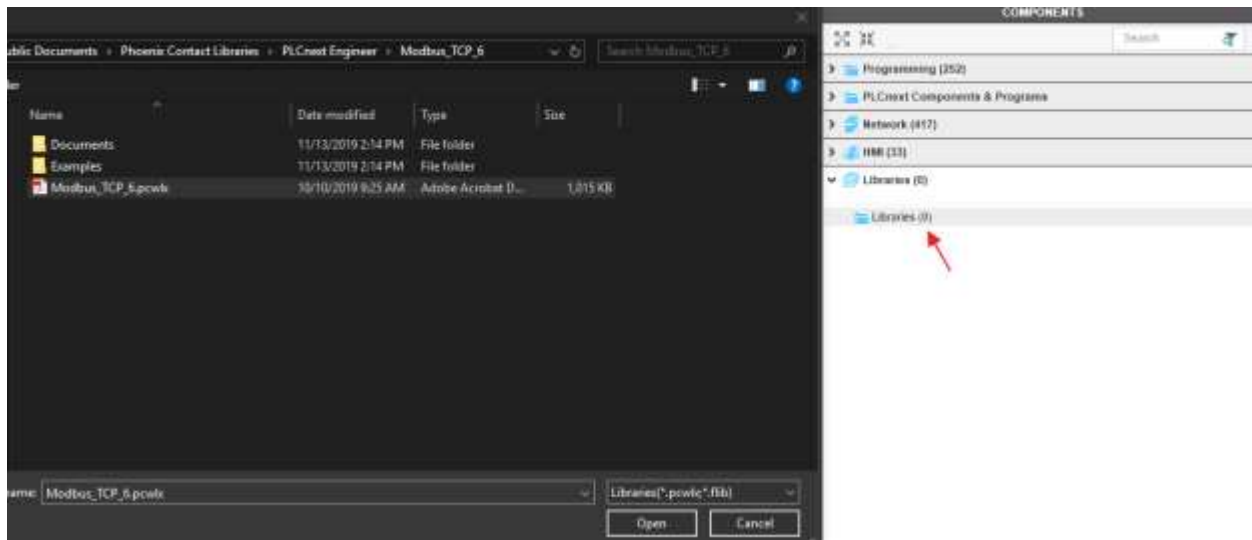
1. Navigate to the PLCNext store - <https://www.plcnextstore.com/#/>. In the search bar on the top right of the page, search for Modbus\_TCP. Once the Modbus\_TCP library is shown, press the download button.
2. Once downloaded, locate the .msi and “Intall.” It is advisable to save all PLCnext libraries in the same file location for simplicity. Follow the prompts until the initial setup is finished.



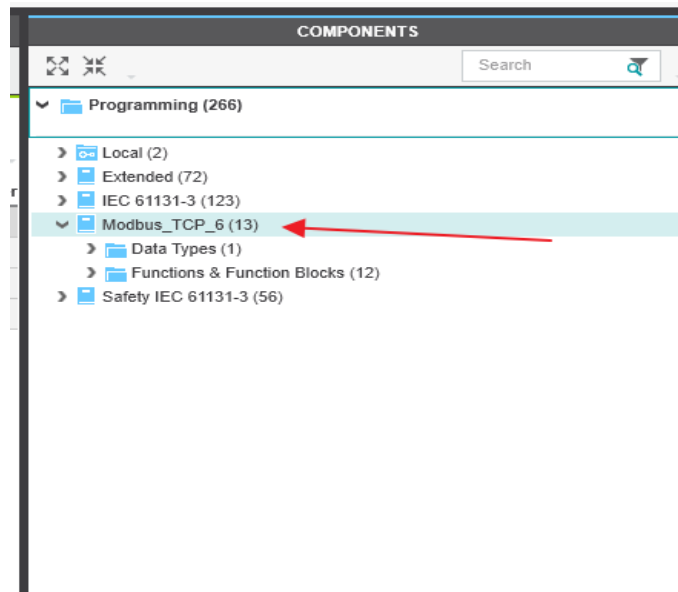
### B. Adding the Modbus TCP library to your PLCnext program.

1. In the Components area, expand the Libraries' tab. Right click on the Libraries option underneath that and select Add User Library. This will open the file explorer. Navigate to the folder that the Modbus library story is stored at and double click on the Modbus\_TCP\_8.pcw\lx folder. (NOTE: The 6 might differ depending on what version of the library you have).

## Setting the AXC F 2152 as a Modbus Client



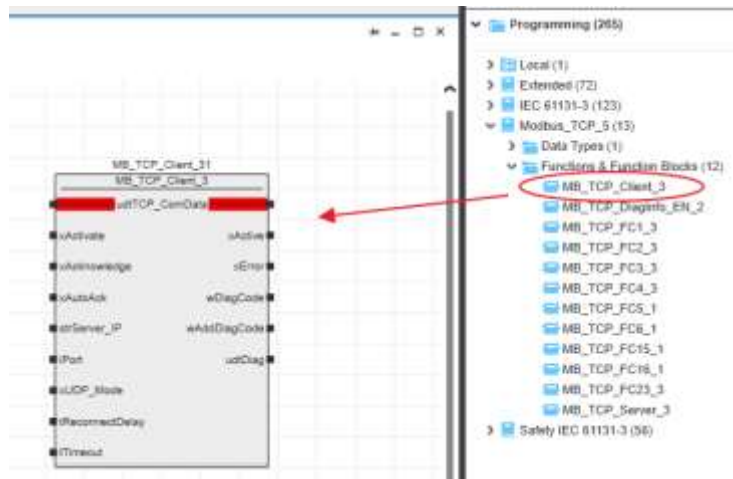
2. To verify that the library was properly inserted into the program, there should now be a Modbus\_TCP\_8 folder in the Components area under the Programming tab.



# Setting the AXC F 2152 as a Modbus Client

## C. Adding the Modbus function blocks.

1. The Modbus\_TCP\_Client function block must be used to configure the AXC F 2152 as a Modbus Client. The function block can be added by expanding the Functions & Function Blocks and dragging the MB\_TCP\_Client function block into your code.



2. Once the function block is placed into the code, variables will need to be assigned to the inputs and outputs. The help file explains the parameters for this function block.

### Input parameters

Name	Type	Description
xActivate	BOOL	Rising edge: Activates the function block. FALSE: Deactivates the function block.
xAcknowledge	BOOL	Rising edge: Error messages are deleted. The function block is not re-initialized.
xAutoAck	BOOL	Automatic acknowledgment of errors. TRUE: The error is acknowledged automatically. Error data is present at the xError, wDiagCode and wAddDiagCode outputs for only one cycle.
strServer_IP	STRING	IP address of the Modbus server. The address should be specified as follows: xxx.xxx.xxx.xxx
iPort	INT	TCP port of the Modbus server. If no address is specified here, the default address 502 is used.
xUDP_Mode	BOOL	FALSE: Modbus-TCP is used. TRUE: Modbus-UDP is used.
tReconnectDelay	TIME	Delay between two IP connect executions. Needed by PLC. Default value = 500ms.
tTimeout	TIME	Timeout for communication monitoring. The Modbus server must respond to a request within the time specified here, otherwise an error is triggered. If no value is specified here, the block operates with a timeout of 2-seconds.

### Output parameters

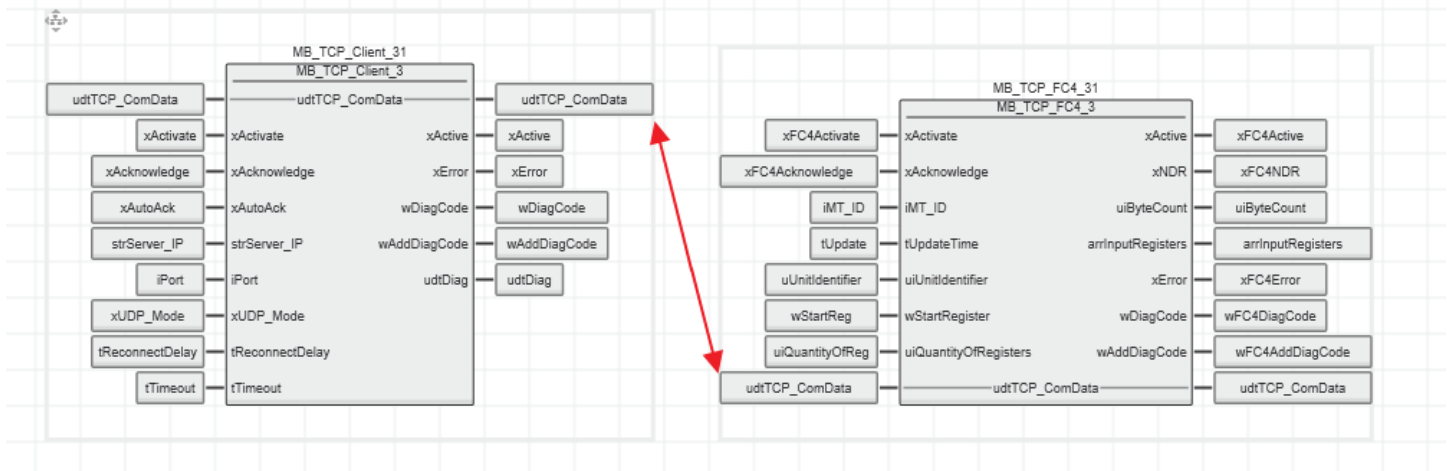
Name	Type	Description
xActive	BOOL	FALSE: Function block is not active. TRUE: Function block is active.
xError	BOOL	TRUE: An error has occurred. For details refer to wDiagCode and wAddDiagCode.
wDiagCode	WORD	Diagnosis code. Refer to diagnostic table.
wAddDiagCode	WORD	Additional diagnosis code. Refer to diagnostic table.
udtDiag	MB_TCP_UDT_CLU_DIAG	Additional diagnostic data.

### Inout parameters

Name	Type	Description
udtTCP_ComData	MB_TCP_UDT_COMMUNICATION	Communication structure of the block family. The blocks are connected with each other by using this structure.

3. For this guide, FC4 will be used to read the digital inputs from the Modbus slave (IL ETH BK DI8 DO4 2TX-PAC – 2703981). To connect the Client Function Block to the FC4 Function block, they need to have the same variable for the inout parameter “udtTCP\_ComData”.

## Setting the AXC F 2152 as a Modbus Client





### D. Running the Project and Pulling Register Data

1. Download the project to the AXC F 2152, and enter debug mode in PLCnext Engineer. Before activating the Client function block, enter the Modbus TCP server's IP address into the "strServer\_IP" variable, and then verify that there is no error after activating it.

Before activating the Function Code function block, enter 1 into the iMT\_ID variable. This input parameter represents the connection between the blocks and may only be assigned once per MB\_TCP\_Client. You will also want to enter the Modbus register of the slave into the "wStartRegister" variable. The input parameter will convert the register into Hexadecimal automatically. Lastly, enter the quantity of registers you want to view into the "uiQuantityOfRegisters".

The (IL ETH BK DI8 DO4 2TX-PAC – 2703981) has an IP address of 192.168.14.174 and the digital input register starts at 8000.




**IL ETH BK DI8 DO4**

- General Instructions
- Device Information
- Device Configuration
- Inline Station
  - Services
  - Process Data Monitoring
  - Remote Diagnostics
  - Bus Configuration
  - PCP Configuration

**IL ETH BK DI8 DO4 2TX-PAC** last update: 15:23:17

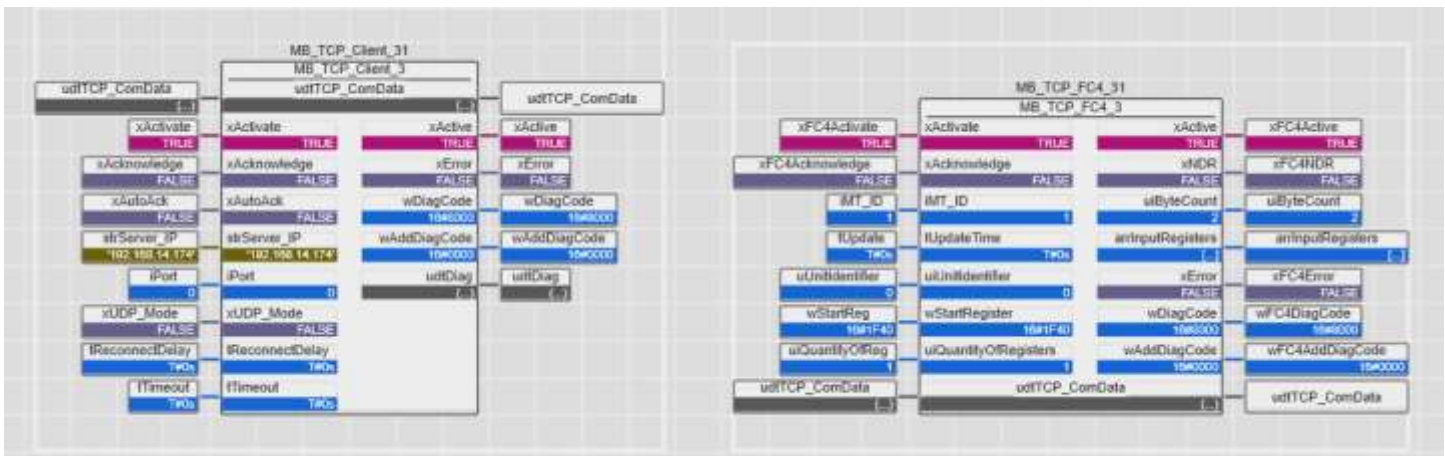
**Bus Configuration**

Baudrate: 2 MBaud

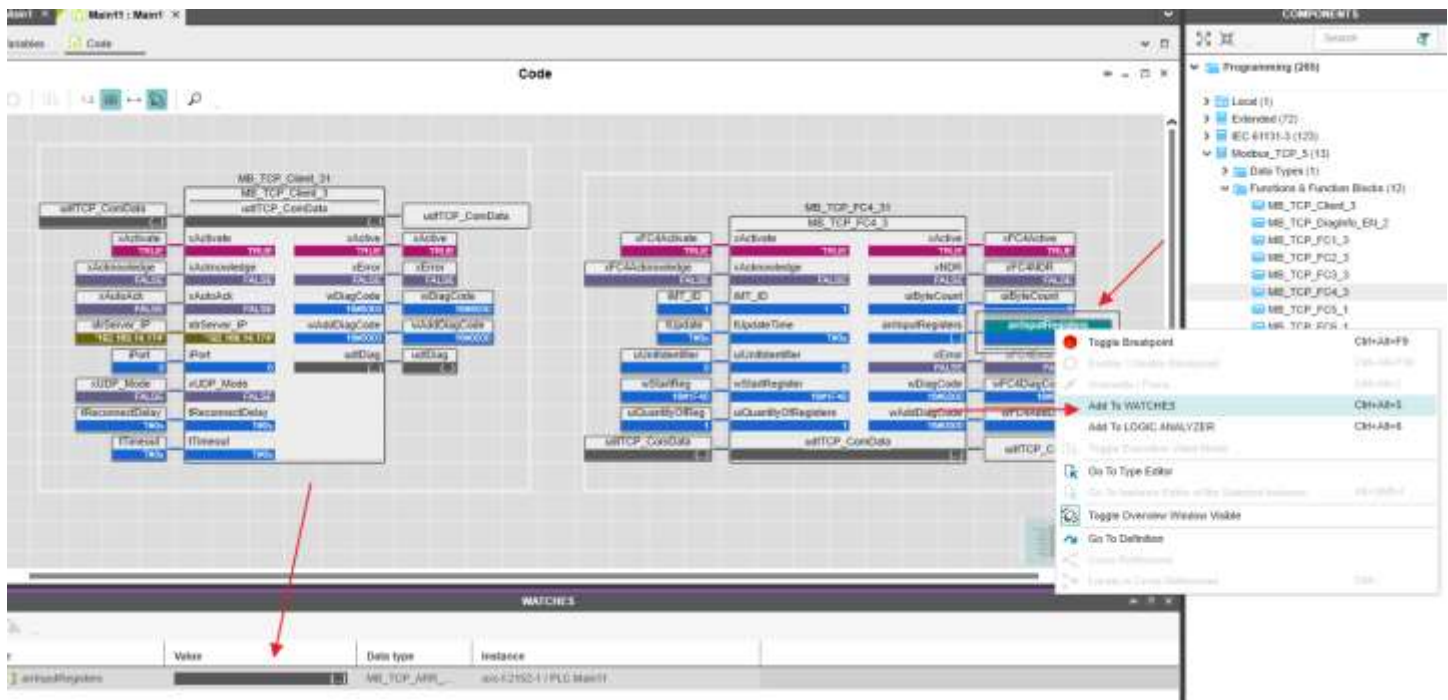
Number	Symbol	Description	Modbus Process Data Register Address	
			IN	OUT
0		IL ETH BK DI8 DO4		
1		Module with 4 digital outputs.	-	8001
2		Module with 8 digital inputs.	8000	-

## Setting the AXC F 2152 as a Modbus Client

2. When everything is connected and the correct parameters are entered, the "xActive" will be true and the "xError" will be false.

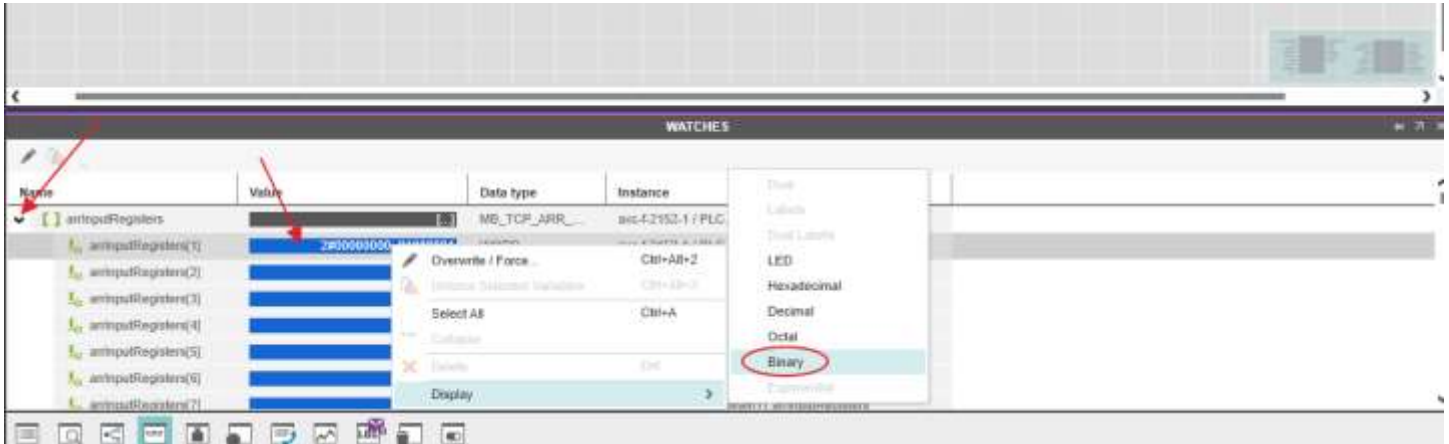


3. To view the data coming from the Modbus slave, right click on the “arrInputRegisters” variable and select “Add to WATCHES”. This will allow you to view that array inside of the watch window.



## Setting the AXC F 2152 as a Modbus Client

- Inside the WATCHES window, click on the arrow to the left of the “arrInputRegisters” to expand the array. The first array, “arrInputRegisters[1]” will represent Modbus register 8000 since that is the value entered into the “wStartRegister”. Right click the value it shows, navigate to “Display”, and select Binary to display this value in Binary.



- Voltage is being applied to digital inputs 1 and 7 on the (IL ETH BK DI8 DO4 2TX-PAC – 2703981). The Binary value from “arrInputRegisters[1]” confirms this by showing bits 0 and 6 high.

[ ]	arrInputRegisters	[...]	MB_
E	arrInputRegisters[1]	2#00000000_01000001	WOF
F	arrInputRegisters[2]	16#0000	WOF