

APPLICATION EXAMPLE

AC500 V3

CI52X-MODTCP IO CONFIGURATION AND COMMUNICAITON VIA BULK DATA MANAGER



Contents

| | |
|---|-----------|
| 1 Disclaimer | 3 |
| 2 Introduction | 4 |
| 2.1 Scope of the document | 4 |
| 2.2 Compatibility | 4 |
| 2.3 Overview | 4 |
| 3 Additional package installation | 5 |
| 3.1 Install High availability Library(PS5601) | 5 |
| 3.2 Install Bulk Data Manager Tool | 5 |
| 3.2.1 BDM Installation | 6 |
| 3.2.2 Problems during BDM installation | 7 |
| 4 Configuration CI52x Cluster | 8 |
| 4.1 IP Configuration | 8 |
| 4.2 Create CI52x Project..... | 9 |
| 4.2.1 Start configuration | 9 |
| 4.2.2 V3 Export..... | 12 |
| 5 PLC AC500 V3 Project..... | 13 |
| 5.1 Automation Builder configuration..... | 13 |
| 5.1.1 CPU Target..... | 13 |
| 5.1.2 Scan Network for Devices | 13 |
| 5.1.3 Add CI52xs library | 14 |
| 5.2 Application Programing | 16 |
| 5.2.1 Import PLCopen xml file into Automation Builder | 16 |
| 5.2.2 User application | 19 |
| 5.3 Task Configuration | 21 |
| 5.4 Program Download | 21 |

1 Disclaimer

A. For customers domiciled outside Germany /

Für Kunden mit Sitz außerhalb Deutschlands

„Warranty, Liability:

The user shall be solely responsible for the use of this products described within this file. ABB shall be under no warranty whatsoever. ABB's liability in connection with application of the products or examples provided or the files included within this products, irrespective of the legal ground, shall be excluded. The exclusion of liability shall not apply in the case of intention or gross negligence. The present declaration shall be governed by and construed in accordance with the laws of Switzerland under exclusion of its conflict of laws rules and of the Vienna Convention on the International Sale of Goods (CISG)."

„Gewährleistung und Haftung:

Der Nutzer ist allein für die Verwendung des in diesem Dokument beschriebenen Produkte und beschriebenen Anwendungsbeispiele verantwortlich.

ABB unterliegt keiner Gewährleistung. Die Haftung von ABB im Zusammenhang mit diesem Anwendungsbeispiel oder den in dieser Datei enthaltenen Dateien - gleich aus welchem Rechtsgrund - ist ausgeschlossen. Dieser Ausschluss gilt nicht im Falle von Vorsatz oder grober Fahrlässigkeit. Diese Erklärung unterliegt Schweizer Recht unter Ausschluss der Verweisungsnormen und des UN-Kaufrechts (CISG)."

B. Nur für Kunden mit Sitz in Deutschland

„Gewährleistung und Haftung:

Die in diesem Dokument beschriebenen Anwendungsbeispiele oder enthaltenen Dateien beschreiben eine mögliche Anwendung der AC500 bzw. zeigen eine mögliche Einsatzart. Sie stellen nur Beispiele für Programmierungen dar, sind aber keine fertigen Lösungen. Eine Gewähr kann nicht übernommen werden.

Der Nutzer ist für die ordnungsgemäße, insbesondere vollständige und fehlerfreie Programmierung der Steuerungen selbst verantwortlich. Im Falle der teilweisen oder ganzen Übernahme der Programmierbeispiele können gegen ABB keine Ansprüche geltend gemacht werden.

Die Haftung von ABB, gleich aus welchem Rechtsgrund, im Zusammenhang mit den Anwendungsbeispielen oder den in dieser Datei enthaltenen Beschreibung wird ausgeschlossen. Der Haftungsausschluss gilt jedoch nicht in Fällen des Vorsatzes, der groben Fahrlässigkeit, bei Ansprüchen nach dem Produkthaftungsgesetz, im Falle der Verletzung des Lebens, des Körpers oder der Gesundheit oder bei schuldhafter Verletzung einer wesentlichen Vertragspflicht. Im Falle der Verletzung einer wesentlichen Vertragspflicht ist die Haftung jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht zugleich ein anderer der in Satz 2 dieses Unterabsatzes erwähnten Fälle gegeben ist. Eine Änderung der Beweislast zum Nachteil des Nutzers ist hiermit nicht verbunden.

Es gilt materielles deutsches Recht unter Ausschluss des UN-Kaufrechts."

2 Introduction

2.1 Scope of the document

The Modbus TCP/IP bus Modules CI521 and CI522 are used as decentralized I/O module in Modbus TCP networks.

This document describes the use of the ABB_CI52x_AC500 library from AC500 HA package for the commutation between CPU (PM5650-2ETH in the example) and CI521 or CI522 modules.



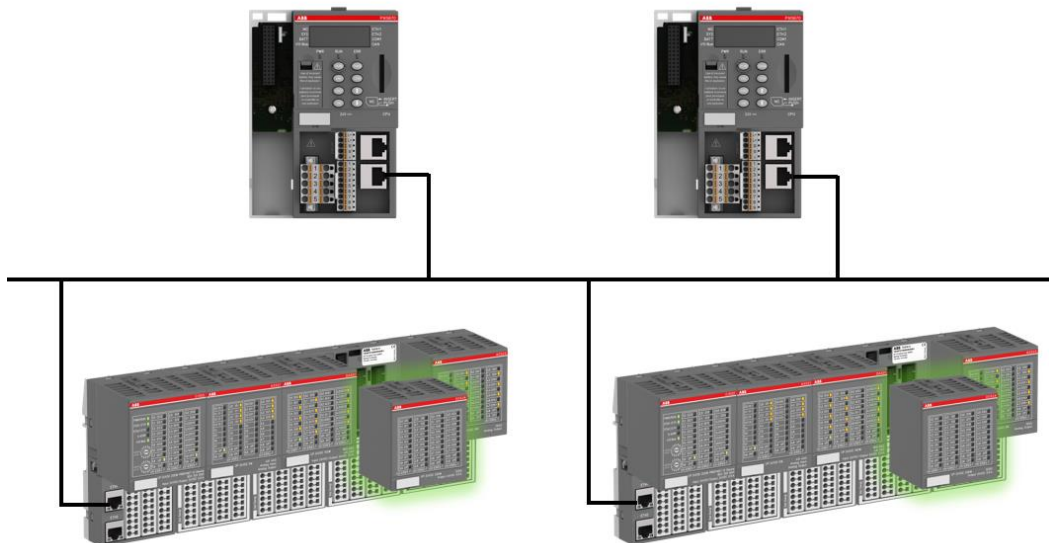
Note: The Function Blocks contained in the library can only be executed in RUN mode of the PLC, but not in simulation mode.

2.2 Compatibility

The application example explained in this document have been used with the below engineering system versions. They should also work with other versions, nevertheless some small adaptations may be necessary, for future versions.

- AC500 V3 PLC
- Automation Builder 2.2.5 or newer
- Bulk Data manager tool
- ABB_CI52x_AC500.compiled-library

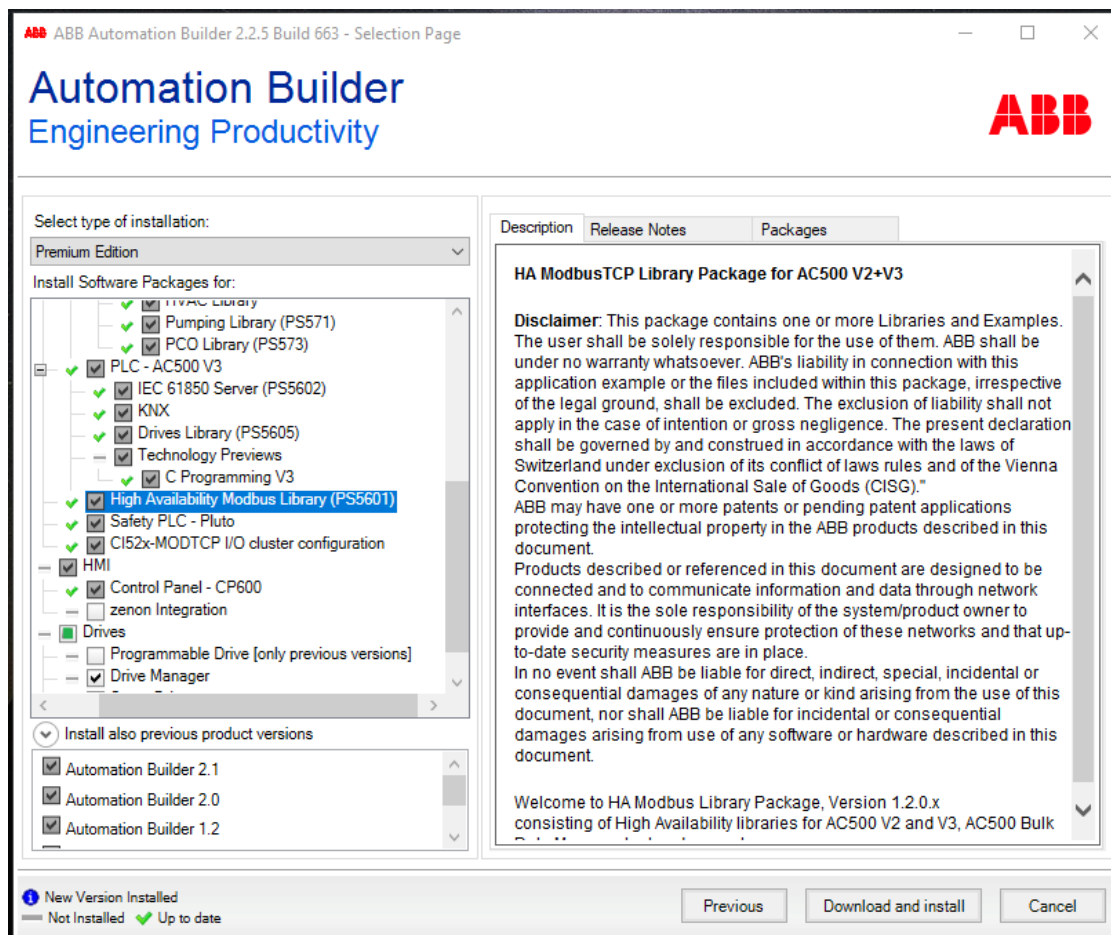
2.3 Overview



3 Additional package installation

3.1 Install High availability Library(PS5601)

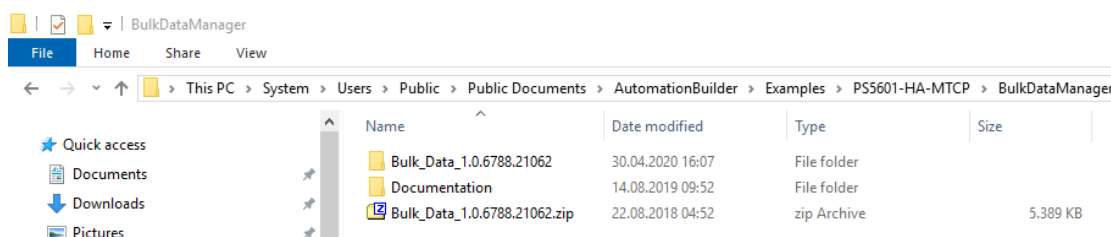
PS5601-High availability Modbus library package can be installed from the 'Automation Builder Installation Manager' by selecting the component.



3.2 Install Bulk Data Manager Tool

After the installation of PS5601-High availability Modbus library package, the installation file of Bulk Data manager Tool can be found in the following folder:

C:\Users\Public\Documents\AutomationBuilder\Examples\PS5601-HA-MTCP\ BulkData-Manager

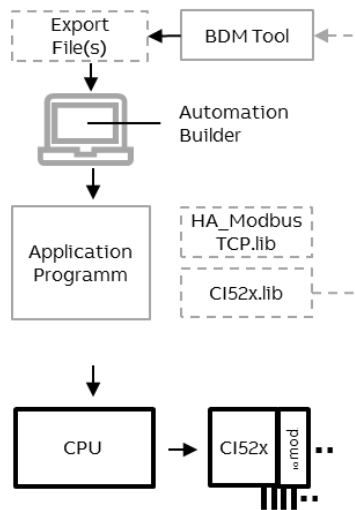


The Bulk Data Manager Tool (BDM) bases on Microsoft Access and helps to manage structured data of larger projects in one place. Typically, engineering is started early and by names and signal lists coming from end-customers or other parties involved in larger projects.

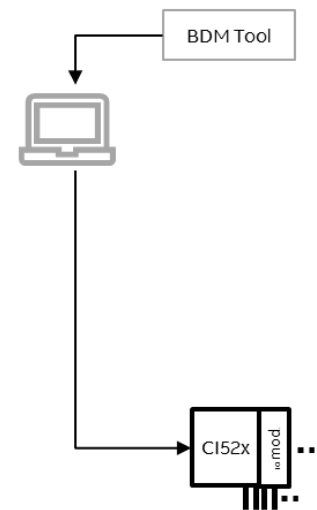
There the BDM tool help significantly to

- Create larger numbers of IO-cluster configurations by selecting and parametrizing suitable AC500 IO Modules,
- Efficiently define variable names in bulk form according to user specifications
- Create suitable code based on function blocks for AC500 - to be imported into Automation Builder (AB)
- In the case of Modbus TCP CI52x library this contains the matching functions for
 - Configuration and parametrization (CI52x modules, IO-modules)
- Also directly via ETH to the modules in case of use direct on other controllers/industrial PCs (right side of Figure).
- Communication
(CI52x communication interfaces ModBusTCP communication to CPU's).

Engineering w. AC500

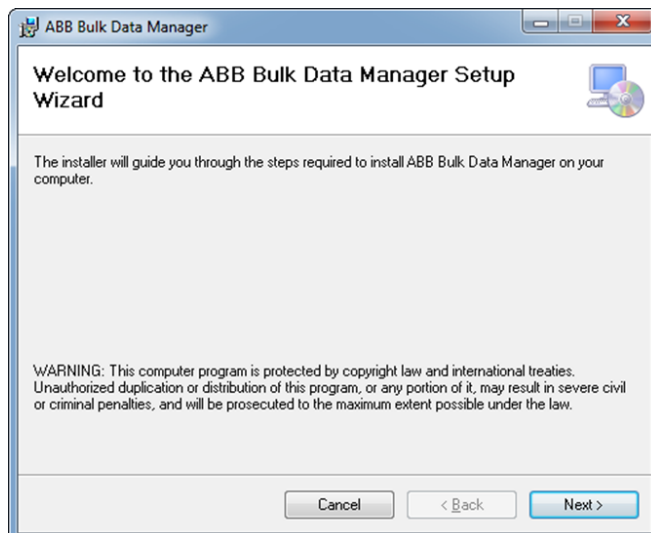


Configuration for other Controller



3.2.1 BDM Installation

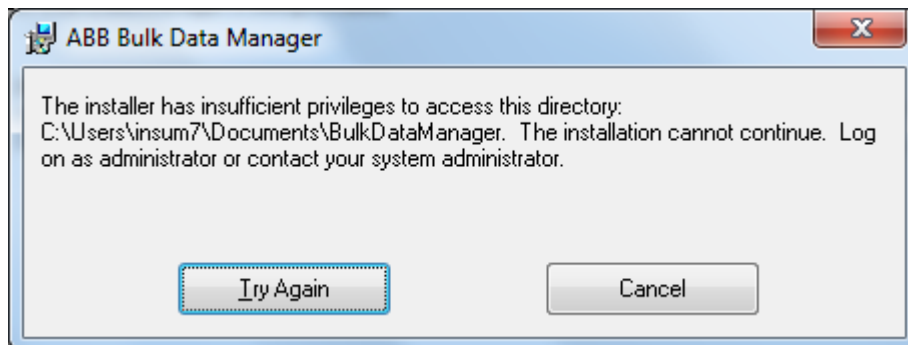
Setup.exe and the *.MSI of the zip file should be in same directory: After launching **setup.exe** the following installation wizard is displayed.



3.2.2 Problems during BDM installation

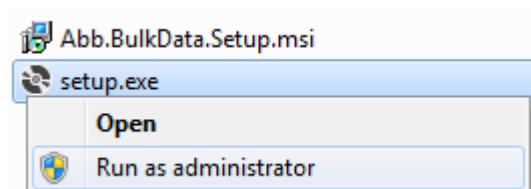
During installation you might face the following problems:

- *The installer has insufficient privileges to access this directory*

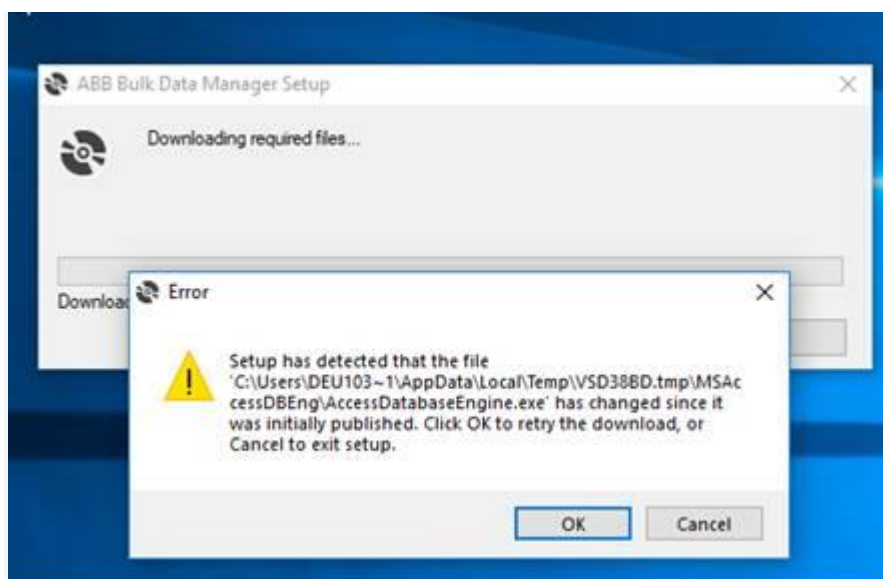


If that is the case, try running the setup as administrator.

Right click setup.exe -> Run as administrator:



- *AccessDatabaseEngine.exe has changed since it was initially published*



The issue is created by the launching of installation of MS access database engine file needed for BDM to run in any machine. This prerequisite file, is triggered to be downloaded from Microsoft website URL (automatically added from visual studio when this file is selected as prerequisite).

In some of the Windows 10 machines, this URL link is broken and Microsoft is not re-directing to the right path. This issue commonly reported in some forums:

<https://developercommunity.visualstudio.com/content/problem/586416/clickonce-bootstrapper-system-no-longer-works-for.html>

Workaround:

1. Instead of the .exe use the .msi file available in the installation folder to install the BDM, which is working. Difference between .exe and .msi files are explained in the link <https://www.symantec.com/connect/articles/understanding-difference-between-exe-and-msi>
2. Manually install the MS access database engine from the location: <http://download.microsoft.com/download/2/4/3/24375141-E08D-4803-AB0E-10F2E3A07AAA/AccessDatabaseEngine.exe> (this link is available in the file package.xml at C:\Program Files (x86)\Microsoft SDKs\ClickOnce Bootstrapper\Packages\MSAccessDBEng)



Note: Link for the MS Access 2013 Runtime:

https://download.microsoft.com/download/5/E/A/5EA6017B-E7FE-40CA-8C3E-57387259F3BF/AccessRuntime_x86_en-us.exe

4 Configuration CI52x Cluster

In this chapter of the example document, steps to create a new configuration CI52x Cluster is explained step by step. As an example to demonstrate the creation of new project, we have considered an example with one CI52c cluster: CI521 with DC523 and AX522 IO modules.

A direct Ethernet connection is required between the PC running Automation Builder and the CI52x-MODTCP module.

4.1 IP Configuration

With the 'IP-Configuration' from the menu 'Tools' in Automation Builder, Users can set the IP address of each components as below. e.g.:

CI521-MODTCP : 192.168.2.100

| Library Manager | | IP-Configuration x | | | | | | |
|-------------------|--------------|---|---------------|-----------|------------|---------------|--------------------|---------------|
| Scan | | <input checked="" type="checkbox"/> ABB Net config protocol (e.g. for AC500 PLC, CI52x-MODTCP devices and ABB Drives) <input type="checkbox"/> Profinet Dynamic Configuration Protocol (DCP) <input type="checkbox"/> EtherCAT | | | | | | |
| (MAC) address | Device | Port | Serial number | Device ID | FW Version | IP Address | Config. IP Address | Device Type |
| 00-24-59-0F-BE-24 | CI521-MODTCP | | 00000150 | 0x02 | -- | 192.168.2.100 | 192.168.2.100 | Modbus Device |
| 00-24-59-0D-0B-8B | PM5650-2ETH | ETH2 | 00000015 | 0x00 | -- | 10.49.123.123 | 10.49.123.123 | AC500 Device |
| 00-24-59-0D-03-6A | PM5650-2ETH | ETH1 | 00000259 | 0x00 | -- | 192.168.22.10 | 192.168.22.10 | AC500 Device |



Note: In the PM5650-2ETH PLC, ETH1 and ETH2 has to be in different sub-net always.

For example, configured IP in this example setup are

PLC : ETH1- 192.168.22.10,255.255.255.0

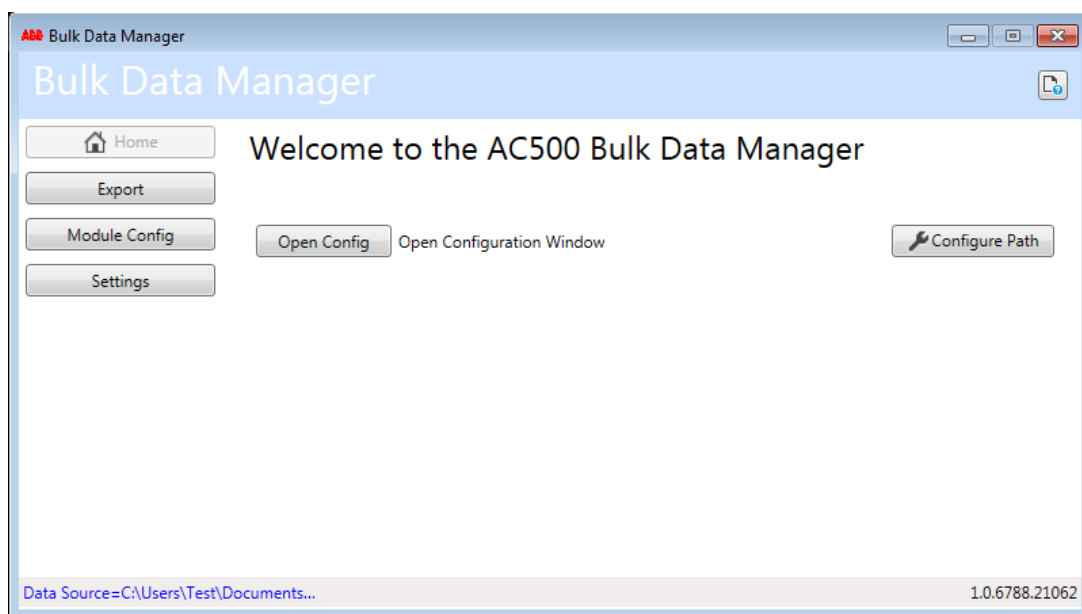
ETH2- 192.168.2.10, 255.255.240.0

4.2 Create CI52x Project

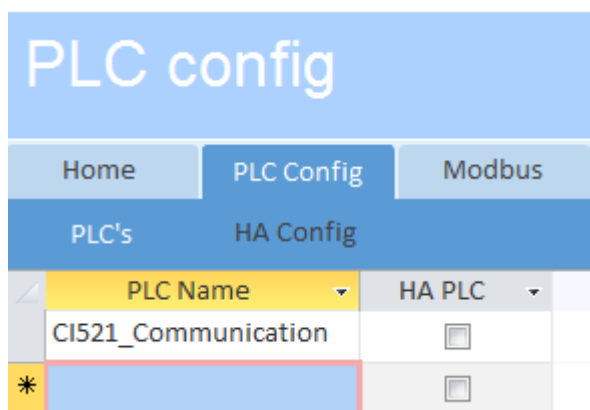
Creation of the cluster configuration using ABB Bulk Data manager tool, importing the configuration to Automation Builder, downloading the project files is covered in the below sections.

4.2.1 Start configuration

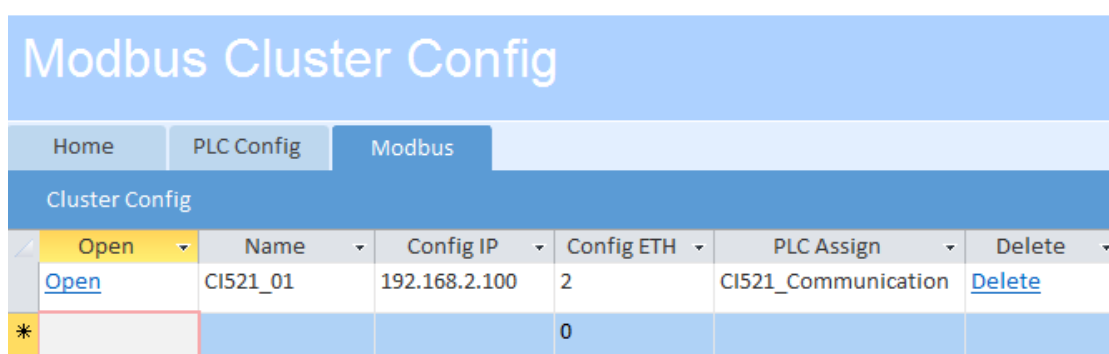
Select 'Open Config', the window 'new Project' appears.



Under "PLC's->PLC Name" enter the name '**CI521_Communication**' and Uncheck 'HA PLC'.



Open the tab 'Modbus', and configure the clusters.



In this table you can add, delete and edit each clusters, including Name, IP address of CI52x modules(e.g. 192.168.2.100 from the example project) and Ethernet index number(if ETH2 used then ETH_Slot=2) which will be used for the communication. To add a new cluster

- Open: Open the configuration of the cluster.
- Name: Unique name of the cluster according to naming rules.
- IP: IP address for CI52x.
- ETH Slot: Ethernet index number (e.g. 1 or 2).
- PLC Assign: Default PLC Name assign
- Delete: Delete cluster

By Clicking on the option 'Open', configure the IO modules attached to CI52xs in the cluster.

Cluster CI521_01

| Config | Module 0 | Module 1 | Module 2 | Module 3 | Module 4 | Module 5 | Module 6 | Module 7 | Module 8 | Module 9 | Module 10 | show all | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|----------|------|------|-----------------|---------------|-----------------|---------------|-----------------|---------------|--------------------|---------------------|--------------------|---------------------|--------------------|---------------------|--------------------|---------------------|--------------------|---------------------|--------------------|---------------------|--------------------|---------------------|---------------------|---------------------|
| <div> <div> <h3>Module Configuration</h3> <table> <thead> <tr> <th>Type</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>Module 0: CI521</td> <td>CL01_00_CI521</td> </tr> <tr> <td>Module 1: DC523</td> <td>CL01_01_DC523</td> </tr> <tr> <td>Module 2: AX522</td> <td>CL01_02_AX522</td> </tr> <tr> <td>Module 3: Not used</td> <td>Enter a module name</td> </tr> <tr> <td>Module 4: Not used</td> <td>Enter a module name</td> </tr> <tr> <td>Module 5: Not used</td> <td>Enter a module name</td> </tr> <tr> <td>Module 6: Not used</td> <td>Enter a module name</td> </tr> <tr> <td>Module 7: Not used</td> <td>Enter a module name</td> </tr> <tr> <td>Module 8: Not used</td> <td>Enter a module name</td> </tr> <tr> <td>Module 9: Not used</td> <td>Enter a module name</td> </tr> <tr> <td>Module 10: Not used</td> <td>Enter a module name</td> </tr> </tbody> </table> </div> <div> <h3>Current Cluster Configuration</h3> <p>Module Group: CI521_01</p> <p>IP Address: 192.168.2.100</p> <p>Config ETH: 2</p> </div> </div> | | | | | | | | | | | | | Type | Name | Module 0: CI521 | CL01_00_CI521 | Module 1: DC523 | CL01_01_DC523 | Module 2: AX522 | CL01_02_AX522 | Module 3: Not used | Enter a module name | Module 4: Not used | Enter a module name | Module 5: Not used | Enter a module name | Module 6: Not used | Enter a module name | Module 7: Not used | Enter a module name | Module 8: Not used | Enter a module name | Module 9: Not used | Enter a module name | Module 10: Not used | Enter a module name |
| Type | Name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 0: CI521 | CL01_00_CI521 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 1: DC523 | CL01_01_DC523 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 2: AX522 | CL01_02_AX522 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 3: Not used | Enter a module name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 4: Not used | Enter a module name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 5: Not used | Enter a module name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 6: Not used | Enter a module name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 7: Not used | Enter a module name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 8: Not used | Enter a module name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 9: Not used | Enter a module name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Module 10: Not used | Enter a module name | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

The module type has to be selected via a combo box. For each module selected, the corresponding tab for editing parameters and IO mapping is enabled. In the example we are using combination CI521+DC523+AX522 for the Cluster.

| Cluster CI521_01 | | | | | | | | | | | | |
|------------------|----------|------------|------------------------------------|---|----------|--|----------|----------|----------|----------|----------|-----------|
| | Config | Module 0 | Module 1 | Module 2 | Module 3 | Module 4 | Module 5 | Module 6 | Module 7 | Module 8 | Module 9 | Module 10 |
| Parameter | Group ID | Module Nur | Parameter | Setting | Edit | | | | | | | |
| IO Mapping | CI521_01 | 0 | Ignore Modul | No | Edit | This parameter allows to set whether the I/O device specified in the PLC | | | | | | |
| Analog Scaling | CI521_01 | 0 | Check supply | On | Edit | Check supply | | | | | | |
| | CI521_01 | 0 | Input delay | 0.1 ms | Edit | Input delay of digital inputs | | | | | | |
| | CI521_01 | 0 | Fast counter | No counter | Edit | Operating mode fast counter | | | | | | |
| | CI521_01 | 0 | Module ID High Byte | 28 | | | | | | | | |
| | CI521_01 | 0 | Module ID Low Byte | 232 | | | | | | | | |
| | CI521_01 | 0 | Error LED | On + failsafe | Edit | Error LED / Failsafe function | | | | | | |
| | CI521_01 | 0 | Timeout for Bus supervision | 50 | Edit | Timeout = value * 10 ms | | | | | | |
| | CI521_01 | 0 | IO Mapping Structure | Dynamic Mapping | | | | | | | | |
| | CI521_01 | 0 | Detect short circuit at outputs | On | Edit | | | | | | | |
| | CI521_01 | 0 | Behaviour DO at comm. Error | Off | Edit | | | | | | | |
| | CI521_01 | 0 | Substitute value binary outputs | 0 | Edit | | | | | | | |
| | CI521_01 | 0 | Detect voltage overflow at outputs | On | Edit | | | | | | | |
| | CI521_01 | 0 | Behaviour AO at comm. Error | Off | Edit | | | | | | | |
| | CI521_01 | 0 | Input 0, channel configuration | Not used | Edit | | | | | | | |
| | CI521_01 | 0 | Input 0, check channel | Plausib(ility), Cut wire, Short circuit | Edit | | | | | | | |
| | CI521_01 | 0 | Input 1, channel configuration | Not used | Edit | | | | | | | |
| | CI521_01 | 0 | Input 1, check channel | Plausib(ility), Cut wire, Short circuit | Edit | | | | | | | |
| | CI521_01 | 0 | Input 2, channel configuration | Not used | Edit | | | | | | | |
| | CI521_01 | 0 | Input 2, check channel | Plausib(ility), Cut wire, Short circuit | Edit | | | | | | | |
| | CI521_01 | 0 | Input 3, channel configuration | Not used | Edit | | | | | | | |
| | CI521_01 | 0 | Input 3, check channel | Plausib(ility), Cut wire, Short circuit | Edit | | | | | | | |
| | CI521_01 | 0 | Output 0, channel configuration | -10 V...+10 V | Edit | | | | | | | |
| | CI521_01 | 0 | Output 0, check channel | Plausib(ility), Cut wire, Short circuit | Edit | | | | | | | |
| | CI521_01 | 0 | Output 1, channel configuration | Not used | Edit | Analog output 1 - Configuration of analog output channel | | | | | | |
| | CI521_01 | 0 | Output 1, check channel | Plausib(ility), Cut wire, Short circuit | Edit | Analog output 1 - Check channel | | | | | | |
| | CI521_01 | 0 | Master IP BYTE 0 | 0 | Edit | | | | | | | |

Choose Setting

Setting

ABB

Not used

10 V...+10 V

0...20 mA

4...20 mA

Configure the clusters' configurations in the Tab 'Module 0' parameters. Settings according to Automation Builder.

- Parameter: description of the parameters
- Setting: actual value of the parameters
- Edit: change value
- Description: additional information

By clicking on 'Edit', the values may be changed. If there is no 'Edit', the value is fixed.

| Cluster CI521_01 | | | | | | | | | | | | |
|------------------|------------|----------|------------|----------|-------------------|------------|--------------------|----------|-------------|----------|----------|-----------|
| | Config | Module 0 | Module 1 | Module 2 | Module 3 | Module 4 | Module 5 | Module 6 | Module 7 | Module 8 | Module 9 | Module 10 |
| Parameter | Group Name | Modu | Module Typ | Termir | Channel | KKS Number | Equipment Unit Key | Signal | Descriptor | | | |
| IO Mapping | CI521_01 | 0 | CI521 | 1.0 | Analog input I0+ | | | | | | | |
| Analog Scaling | CI521_01 | 0 | CI521 | 1.1 | Analog input I1+ | | | | | | | |
| | CI521_01 | 0 | CI521 | 1.2 | Analog input I2+ | | | | | | | |
| | CI521_01 | 0 | CI521 | 1.3 | Analog input I3+ | | | | | | | |
| | CI521_01 | 0 | CI521 | 1.5 | Analog output O0+ | AIO | DE | AO0 | AIO Example | | | |
| | CI521_01 | 0 | CI521 | 1.6 | Analog output O1+ | | | | | | | |
| | CI521_01 | 0 | CI521 | 2.0 | Digital input I0 | | | | | | | |
| | CI521_01 | 0 | CI521 | 2.1 | Digital input I1 | | | | | | | |
| | CI521_01 | 0 | CI521 | 2.2 | Digital input I2 | | | | | | | |
| | CI521_01 | 0 | CI521 | 2.3 | Digital input I3 | | | | | | | |
| | CI521_01 | 0 | CI521 | 2.4 | Digital input I4 | | | | | | | |
| | CI521_01 | 0 | CI521 | 2.5 | Digital input I5 | | | | | | | |
| | CI521_01 | 0 | CI521 | 2.6 | Digital input I6 | | | | | | | |
| | CI521_01 | 0 | CI521 | 2.7 | Digital input I7 | | | | | | | |
| | CI521_01 | 0 | CI521 | 3.0 | Digital output D0 | DIO | DE | DO0 | DIO Example | | | |
| | CI521_01 | 0 | CI521 | 3.1 | Digital output D1 | | | | | | | |
| | CI521_01 | 0 | CI521 | 3.2 | Digital output D2 | | | | | | | |
| | CI521_01 | 0 | CI521 | 3.3 | Digital output D3 | | | | | | | |
| | CI521_01 | 0 | CI521 | 3.4 | Digital output D4 | | | | | | | |
| | CI521_01 | 0 | CI521 | 3.5 | Digital output D5 | | | | | | | |
| | CI521_01 | 0 | CI521 | 3.6 | Digital output D6 | | | | | | | |
| | CI521_01 | 0 | CI521 | 3.7 | Digital output D7 | | | | | | | |

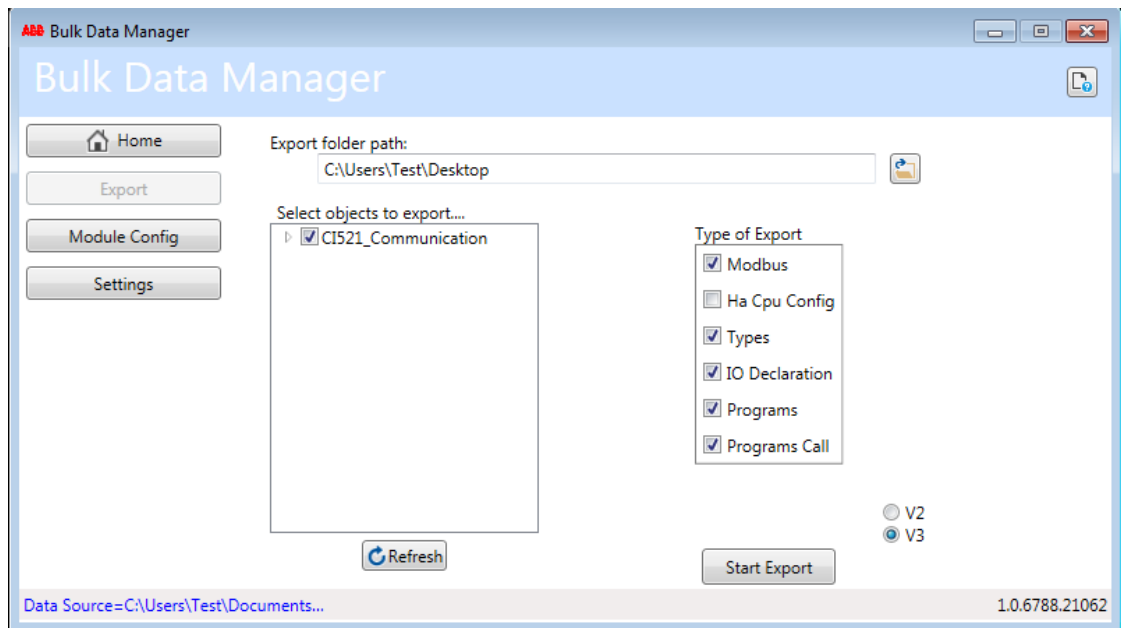
IO mapping for the inputs and outputs on the CI52x module to be configured under the IO mapping tab. In this table all available I/O channels can easily be assigned to a variable.



Note: The IO mapping will be imported into Global Variables type's area automatically which can be called easily in the project.

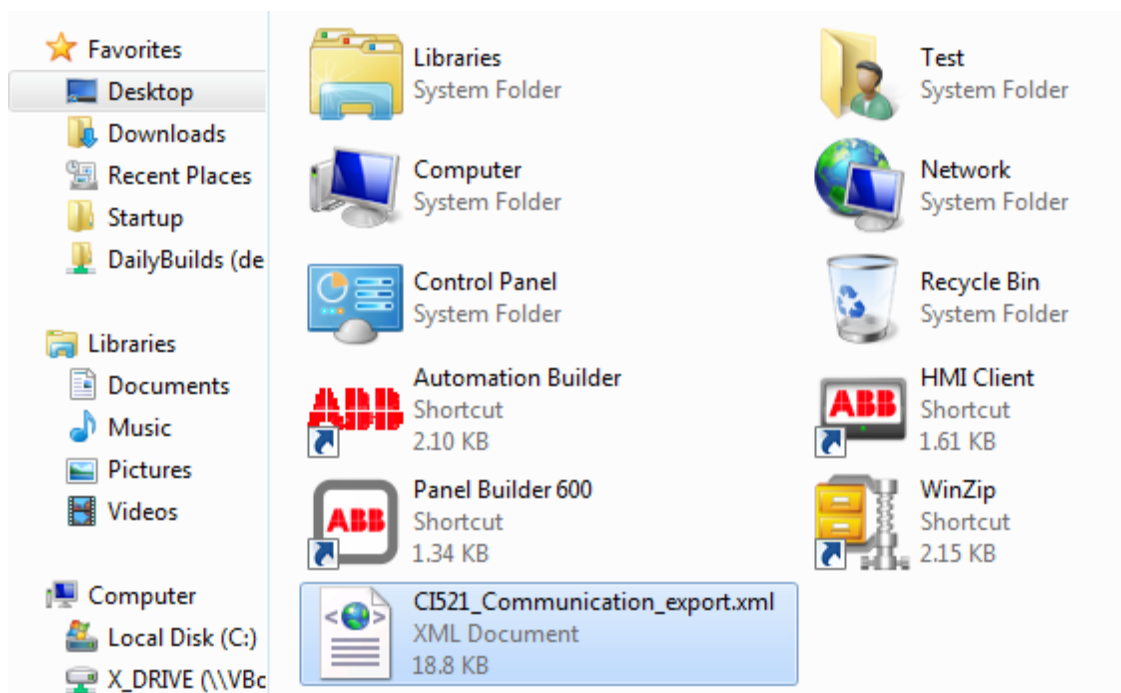
4.2.2 V3 Export

Once the configuration is done, close the access tool and in the Bulk Data Manager popup window and select the export option as shown below.



Check all options in the 'Type of Export' except for 'Ha Cpu Config', so that you will export complete cluster configuration.

Select the desired path where you would like to store the exported xxx.xml file.



The export file is in the desired path, which will be imported into the Automation Builder.



Note: Do remember to press Button 'Refresh' to update the recent changes into the export window.

5 PLC AC500 V3 Project

This chapter describes how CI52x function block is used for the Modbus TCP communication program.

5.1 Automation Builder configuration

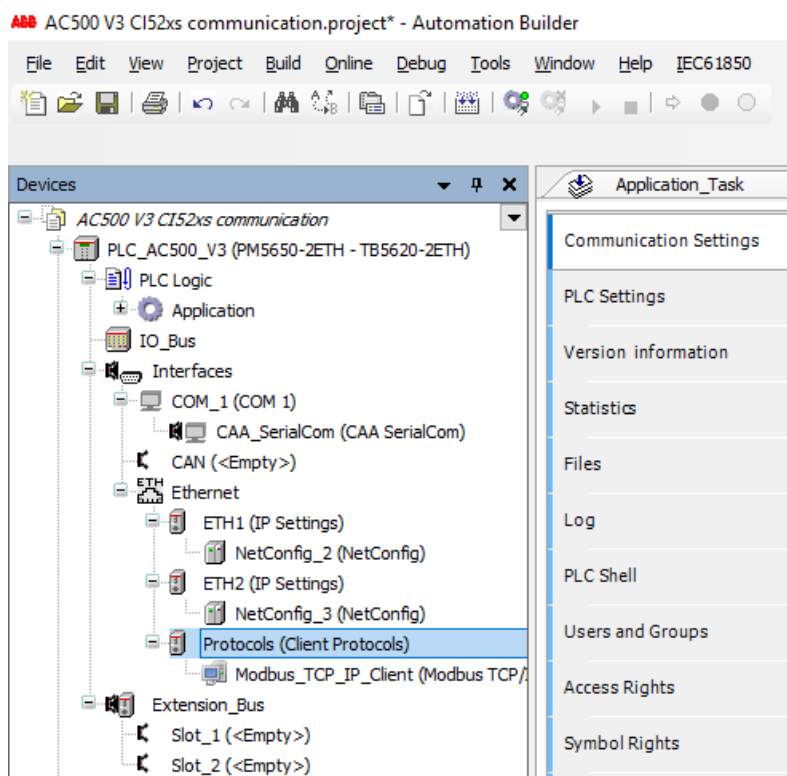


Note: User can skip the following steps by opening the provided example project directly.

5.1.1 CPU Target

Right click in the device tree on the root of the 'Project' and click 'Add Object'.

Select PLC 'PM5650-2ETH' and click 'Add object', the node 'PLC_AC500_V3' appears in your device tree.



Right click on "Protocols" node under "Ethernet" node and select 'Add object' 'Modbus_TCP_IP_Client'. This Modbus TCP client configuration is for both ETH1 and ETH2.

To use the Modbus TCP for the communication between PLC and fieldbus module CI52xs, do configure the Modbus TCP client under the Ethernet protocols.

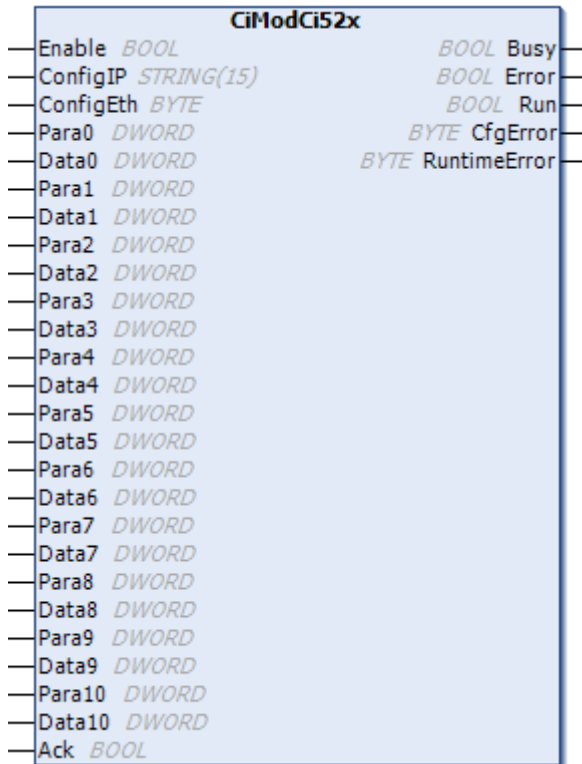
5.1.2 Scan Network for Devices

PLC and CI52xs must be given a fix IP address, with the 'IP-Configuration' from the menu 'Tools' in Automation Builder, Users can set the IP address of CPU as below. e.g.:

| IP-Configuration X | | | | | | | |
|--------------------|-------------|---|---------------|-----------|---------------|--------------------|--------------|
| Scan | | <input checked="" type="checkbox"/> ABB Net config protocol (e.g. for AC500 PLC, CI52x-MODTCP devices and ABB Drives) <input type="checkbox"/> Profinet Dynamic Configuration Protocol (DCP) <input type="checkbox"/> EtherCAT | | | | | |
| (MAC) address | Device | Port | Serial number | Device ID | IP Address † | Config. IP Address | Device Type |
| 00-24-59-0D-03-6A | PM5650-2ETH | ETH1 | 00000259 | 0x00 | 192.168.22.10 | 192.168.22.10 | AC500 Device |
| 00-24-59-0D-03... | PM5650-2ETH | ETH2 | 00000259 | 0x00 | 192.168.2.10 | 192.168.2.10 | AC500 Device |

5.1.3 Add CI52xs library

CI52x library is a part of HA Modbus TCP library package. **CiModCi52x** Function block is to establish the communication between AC500 PLC and Fieldbus Modbus module CI521 or CI522.



Using this function block, also the status of CI52x modules configured can be known.

5.1.3.1 Components of CI52x library

The CI52x library contains the following Function Block, visualization and variables.

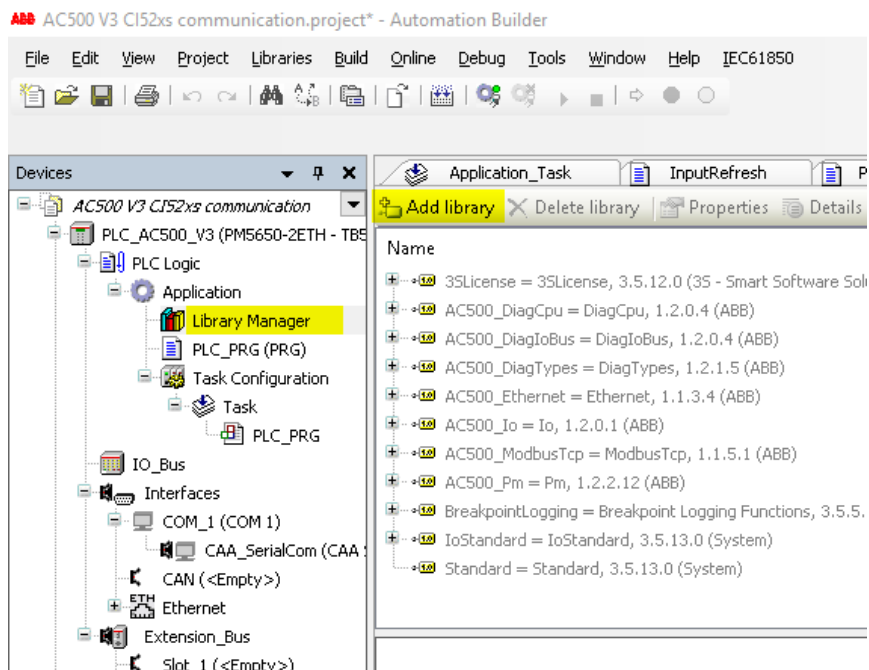
- Function Block
 - CiModCi52x
 - CiModDiag
- Visualization
 - VisuCiModCi52x
 - VisuCiModDiag
- Global Variables
 - CI52x_GLOBAL_INTERNAL
 - CI52x_GLOBAL_VARIABLES



Note: For more details on the Function Block CI52x, refer the online help of Automation Builder 2.2.5 or higher.

5.1.3.2 Library CI52x

To run the example program, we should add the CI52x library installed into the library Manager.



Click on 'Add library' and you will be able to see the library in the path shown in image below, double click on the 'CI5xs' to add it into the library manager.

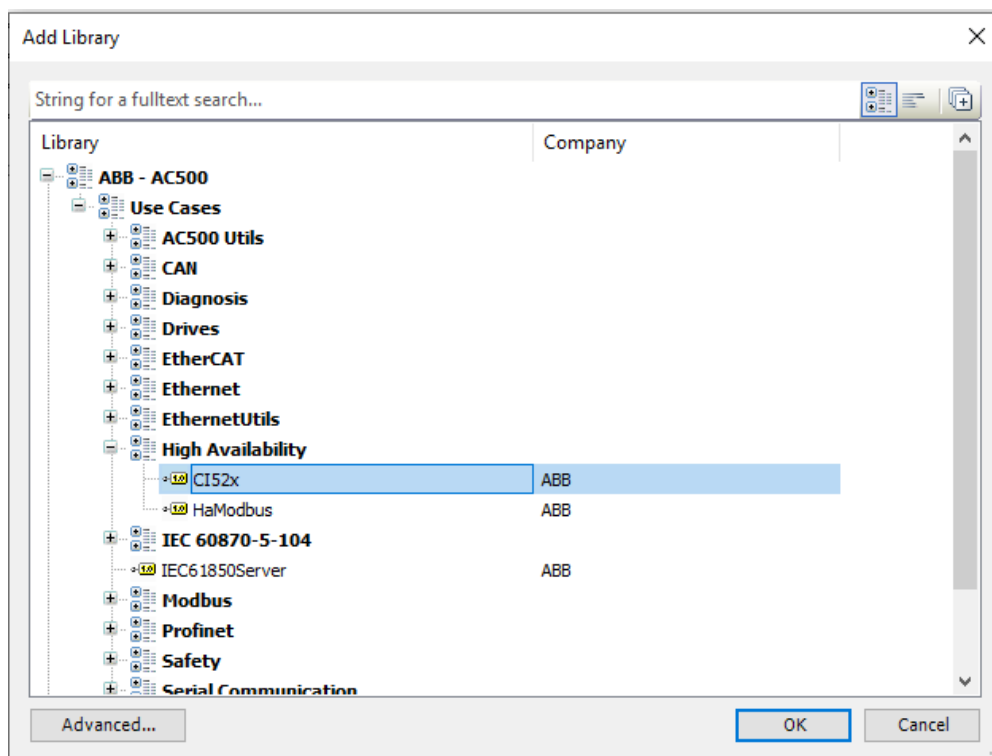


ABB AC500 V3 CI52xs communication.project* - Automation Builder

File Edit View Project Libraries Build Online Debug Tools Window Help IEC61850

Devices

AC500 V3 CI52xs communication

PLC_AC500_V3 (PM5650-2ETH - TB5

PLC Logic

Application

Library Manager

PLC_PRG (PRG)

Task Configuration

Task

PLC_PRG

IO_Bus

Interfaces

COM 1 (COM 1)

Add library X Delete library Properties Details P

Name

- 3SLicense = 3SLicense, 3.5.12.0 (3S - Smart Software Solutions G
- AC500_CI52x = CI52x, 1.2.0.10 (ABB)
- AC500_DiagCpu = DiagCpu, 1.2.0.4 (ABB)
- AC500_DiagIoBus = DiagIoBus, 1.2.0.4 (ABB)
- AC500_DiagTypes = DiagTypes, 1.2.1.5 (ABB)
- AC500_Ethernet = Ethernet, 1.1.3.4 (ABB)
- AC500_Io = Io, 1.2.0.1 (ABB)
- AC500_ModbusTcp = ModbusTcp, 1.1.5.1 (ABB)
- AC500_Pm = Pm, 1.2.2.12 (ABB)

Note: If user have already installed the different version of CI52x library, then user must click on Placeholders and select appropriate version of library if version conflict error occurs.

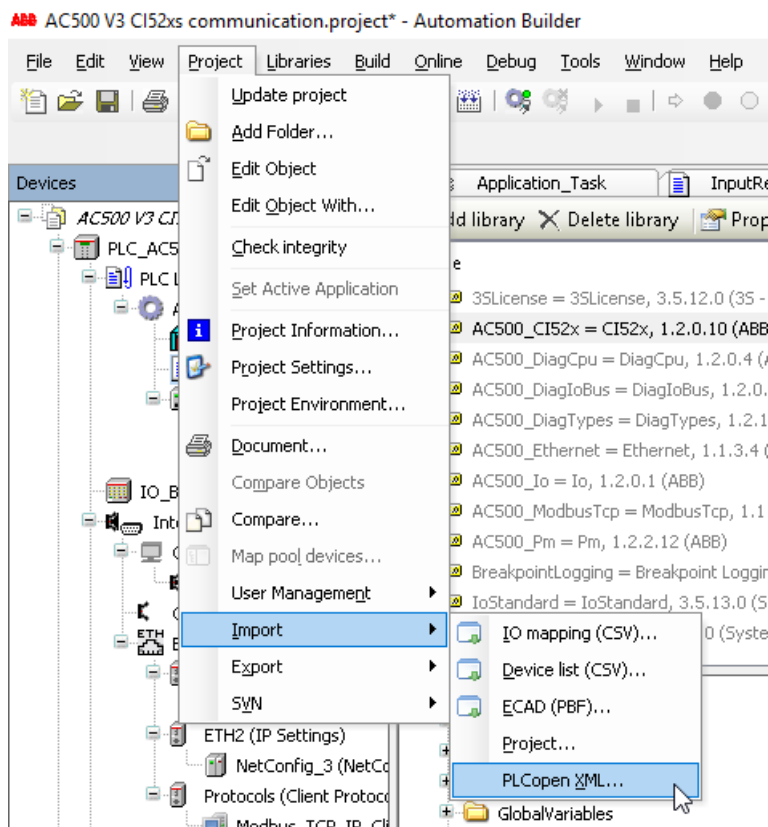
5.2 Application Programming

The export file generated from the Bulk Data Manager Tool is in the PLCopen format (*.xml).

5.2.1 Import PLCopen xml file into Automation Builder

To import the 'CI521_Communication_export.xml' file into Automation Builder with the following steps:

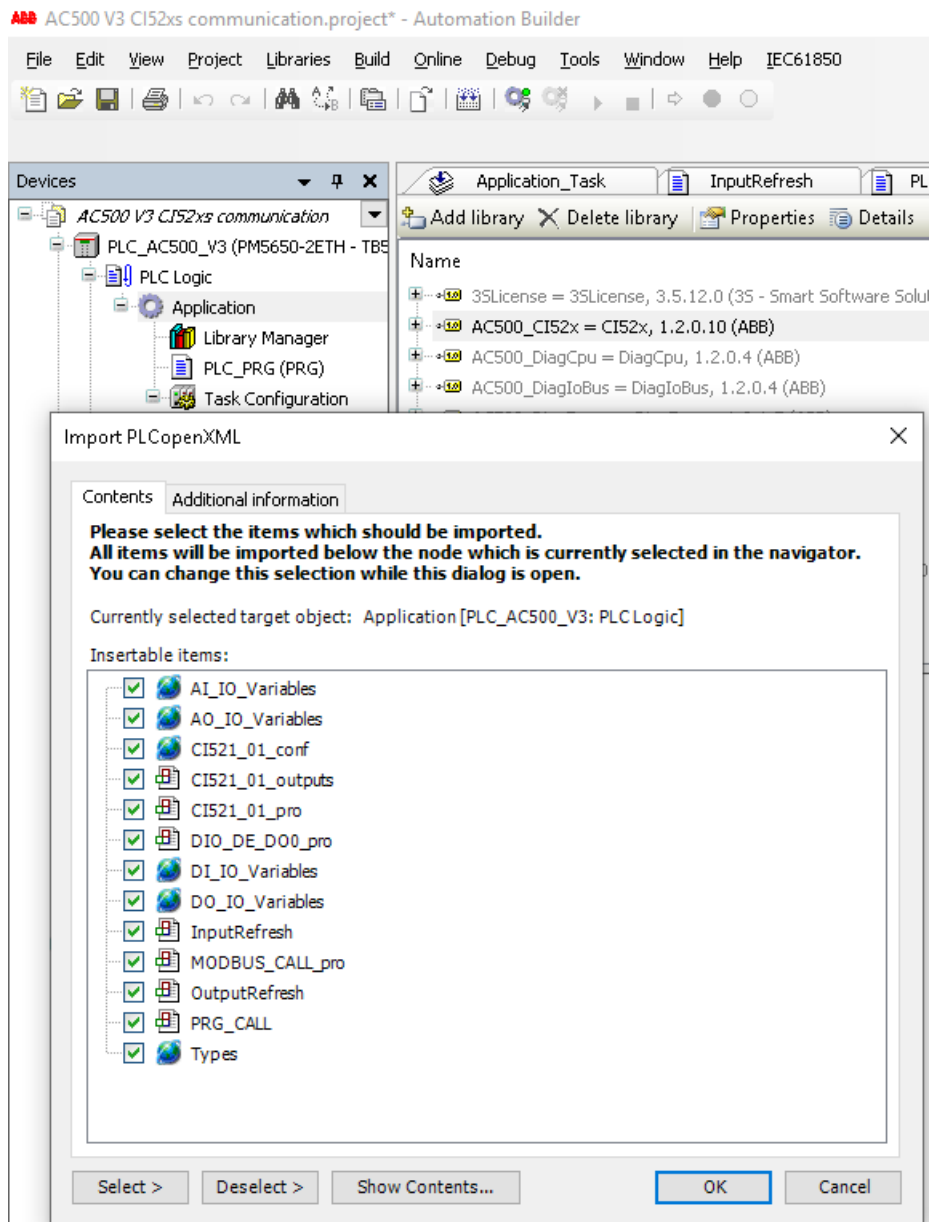
Select Project→ Import→ PLCopen XML...



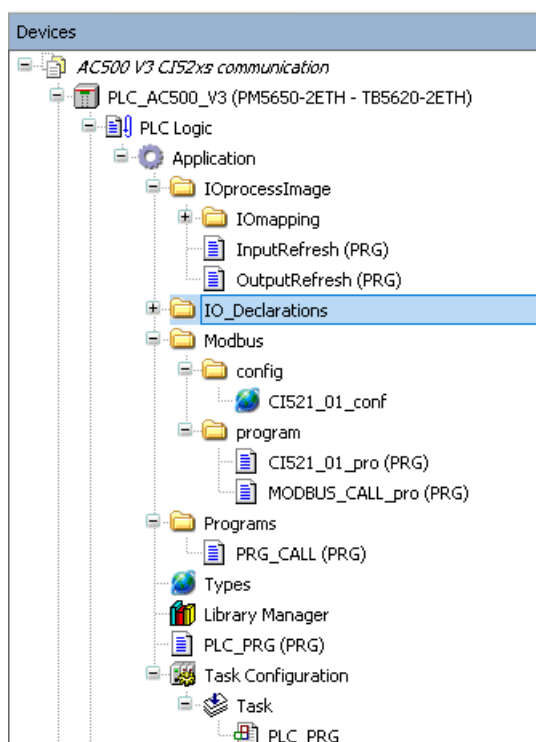
In the file Dialog select the desired export file 'CI521_Communication_export.xml'.

A Dialog will open, where all items contained in the export file are listed.

To select a target object, click on any object under 'Devices' or 'POUs'. In this case 'application' was selected as target object:



Import the POUs, Global variables under the application node of the PLC device tree. e.g. the Objects are imported with the defined folder structure.



Note: the imported codes could be modified if needed. e.g. CI52xs Configuration parameters, IO mapping...

5.2.2 User application

To test the communication, create one POU named 'User_Application_PRG' in the project.

The screenshot displays the SIMATIC Manager interface. On the left, the 'Devices' tree shows the project structure for 'AC500 V3 CI52xs communication'. The 'Application' folder is expanded, showing sub-folders like 'Diagnosis', 'IOprocessImage', 'IO_Declarations', 'Modbus', 'Types', 'Library Manager', 'PLC_PRG (PRG)', 'User_Application_PRG (PRG)', and 'Task Configuration'. The 'User_Application_PRG (PRG)' is selected. On the right, the 'User_Application_PRG' program is shown in the editor. The code is as follows:

```

1  PROGRAM User_Application_PRG
2  VAR
3      Var_Byte: BYTE;
4      Var_Word: WORD;
5  END_VAR

1  IF Var_Byte < 200 THEN
2      Var_Byte:=Var_Byte+5;
3  ELSE
4      Var_Byte := 0;
5  END_IF
6  IF Var_Word < 30000 THEN
7      Var_Word:=Var_Word+159;
8  ELSE
9      Var_Word := 0;
10 END_IF
11 CL01_00_CI521.QB0:=Var_Byte;
12 CL01_01_DC523.QB0:=Var_Byte;
13 CL01_01_DC523.QB1:=Var_Byte;
14 CL01_01_DC523.QB2:=Var_Byte;
15 CL01_02_AX522.QW0:=Var_Word;
16 CL01_02_AX522.QW1:=Var_Word;
17 CL01_02_AX522.QW2:=Var_Word;
18 CL01_02_AX522.QW3:=Var_Word;
19 CL01_02_AX522.QW4:=Var_Word;
20 CL01_02_AX522.QW5:=Var_Word;
21 CL01_02_AX522.QW6:=Var_Word;
22 CL01_02_AX522.QW7:=Var_Word;

```

At the start of the Application task the InputRefresh program has to be called. It copies data from the Modbus via the structure CiModDataxx to the user variables.

The screenshot displays the SIMATIC Manager interface. On the left, the 'Devices' tree shows the project structure for 'AC500 V3 CI52xs communication'. The 'PLC_AC500_V3 (PM5650-2ETH - TB5620-2ETH)' is selected. The 'PLC Logic' folder is expanded, showing sub-folders like 'Application', 'IOprocessImage', 'IOmapping', 'InputRefresh (PRG)', 'OutputRefresh (PRG)', 'IO_Declarations', 'Modbus', 'config', 'program', 'CI521_01_pro (PRG)', 'MODBUS_CALL_pro (PRG)', 'Programs', 'PRG_CALL (PRG)', 'Types', 'Library Manager', 'PLC_PRG (PRG)', 'User_Application_PRG (PRG)', 'Task Configuration', and 'Task'. The 'User_Application_PRG (PRG)' is selected. On the right, the 'PLC_PRG' program is shown in the editor. The code is as follows:

```

1  PROGRAM PLC_PRG
2  VAR
3  END_VAR

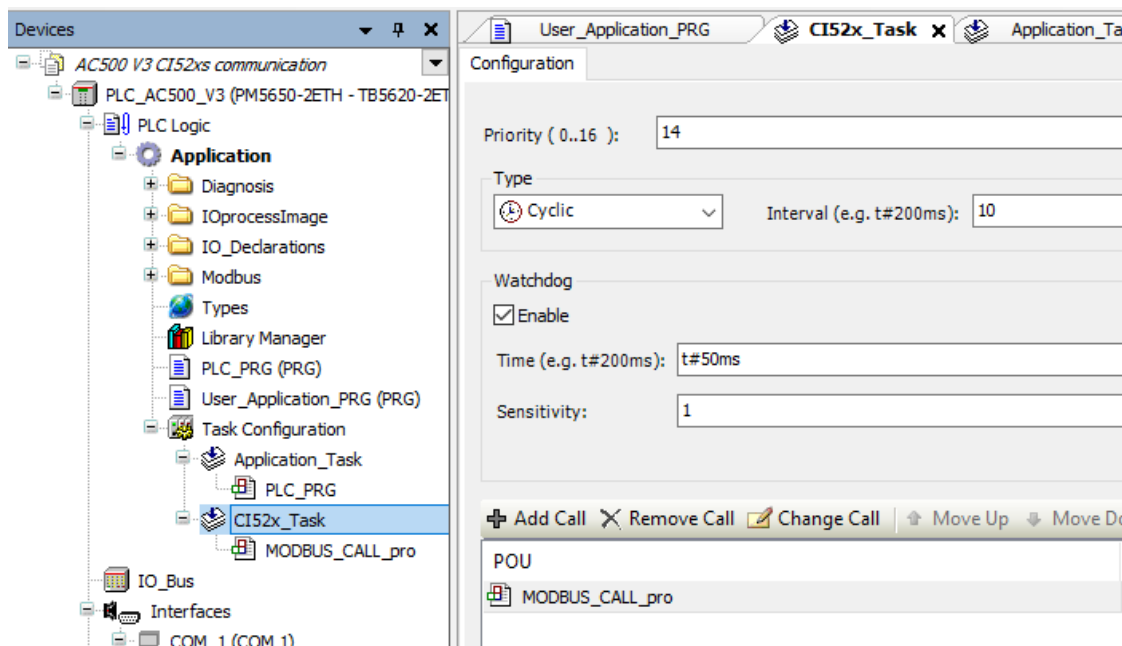
1  InputRefresh();
2  User_Application_PRG();
3  OutputRefresh();

```

OutputRefresh program is called as a last step. It copies data from variables via structure CiModDataxx to Modbus

5.3 Task Configuration

Once the program is finished, tasks must be configured in the task configuration.



Add a task under Task configuration nodes, configure separate task to Application and Modbus communication with necessary cycle time and Priority. In the example project, Modbus 'CI52x_task' at T#10ms with a watchdog time of 50ms and Application task is at T# 20ms (setting value affected by cpuload and plcload).



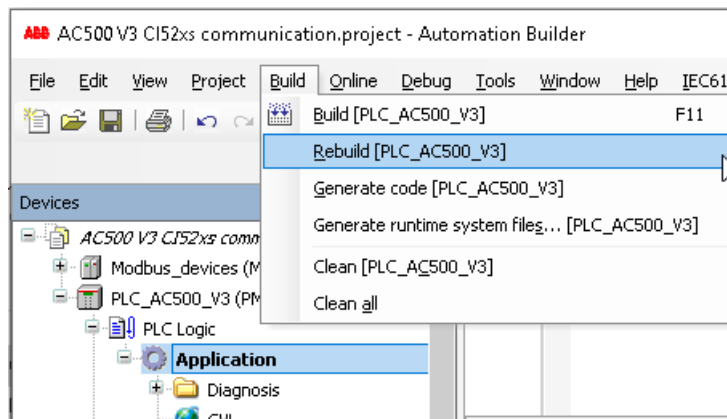
Note:

cpuload: This value represents the time the PLC requires to calculate all processes running on the PLC. For a good system performance this value should be less than 80%. In case of a higher value, the degree of utilization should be reduced by using a more powerful PLC or by reducing the amount of processes.

plcload: This value represents the time the PLC requires to calculate all real time processes. Real time processes are either high priority system tasks or IEC tasks with a priority between 0 and 15. For a good system performance this value should be less than 60%. In case of a higher value, the degree of utilization should be reduced by using a more powerful PLC.

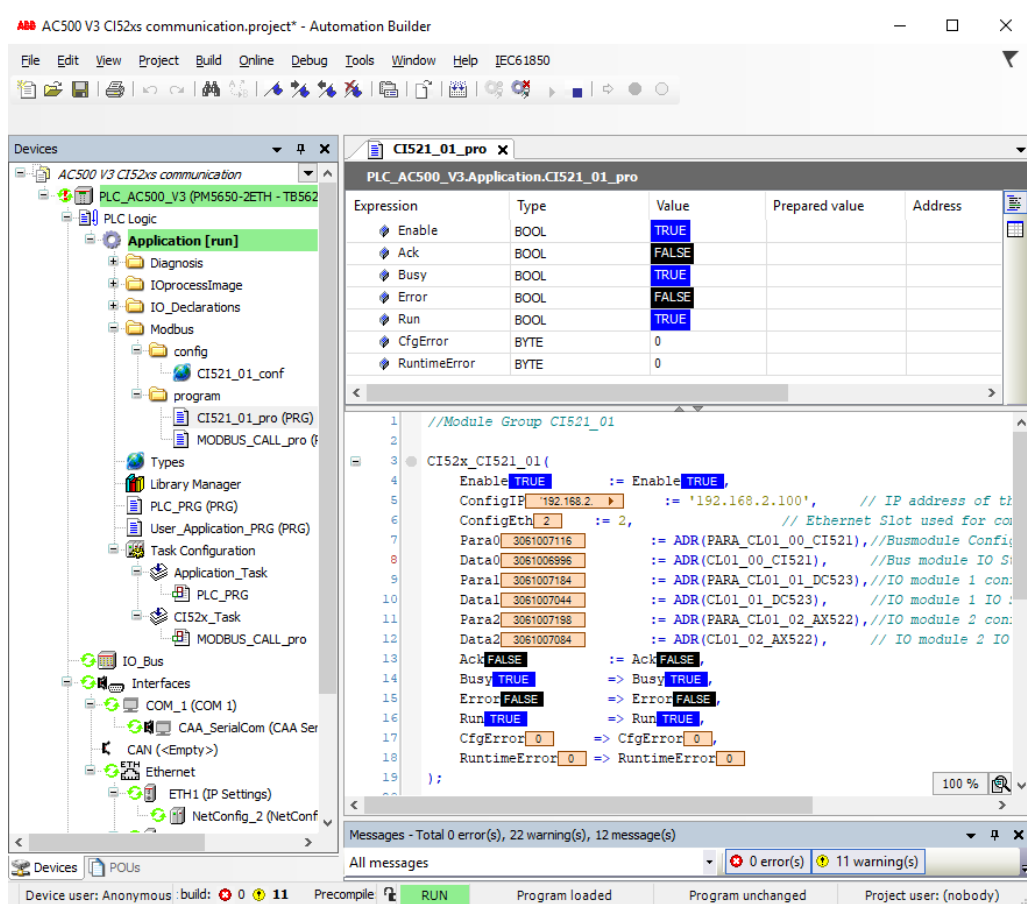
5.4 Program Download

Build the program, to check for any compilation error exists before downloading.



The CPU program can be downloaded to PLC using normal TCP/IP gateway. After the download, create a boot project.

For the best performance, ensure that the online connection is running



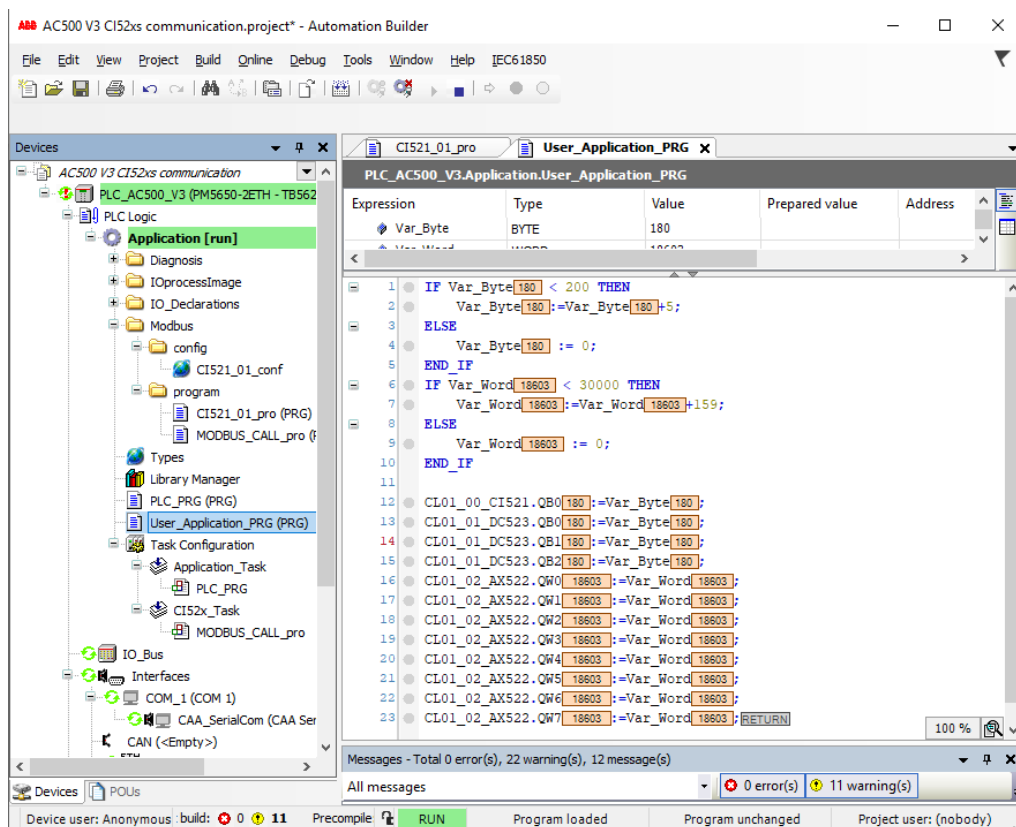


ABB Automation Products GmbH
Eppelheimer Straße 82
69123 Heidelberg, Germany
Phone: +49 62 21 701 1444
Fax: +49 62 21 701 1382
E-Mail: plc.support@de.abb.com
www.abb.com/plc

We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB AG does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB AG.
Copyright© 2020 ABB. All rights reserved