APPLICATION EXAMPLE

CP600 HA V2 CONFIGURATION

NODE OVERRIDE IP VIA GENERIC MODBUS

Contents

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

Currently users have two solutions to realize the communication between CP600 and AC500 HA. One is to switch the connection of CP600 from AC500 V2 HA through CP600's own JavaScript. The advantage of this solution is that in the communication process, CP600 plays a leading role in communication switching. It will determine its communication connection according to the status of Primary CPU. The disadvantage is that users must understand how to use JavaScript language.

Another solution is to implement communication switching through standard Modbus .The advantage of this solution is that during the communication process, the HA CPUs play a leading role in the communication switching with CP600, and the Primary CPU sends its own IP address to CP600 through Modbus, so that CP600 can passively follow the Primary CPU to switch communication connection. It is not necessary for users to understand the JavaScript language of CP600. The disadvantage is that the execution of additional codes will increase the CPU load and the number of sockets.

Both solutions can perfectly realize the communication between the CP600 and AC500 HA. The user can choose one of the solutions to realize the communication connection.

This Application example describes how to switch the connection of CP600 from AC500 V2 HA PLC through Modbus. As for another solution, please refer to the application example <Application Example CP600 HA V2 configuration - Node Override IP via JavaScript> for details.

Switching can be trigger in different ways:

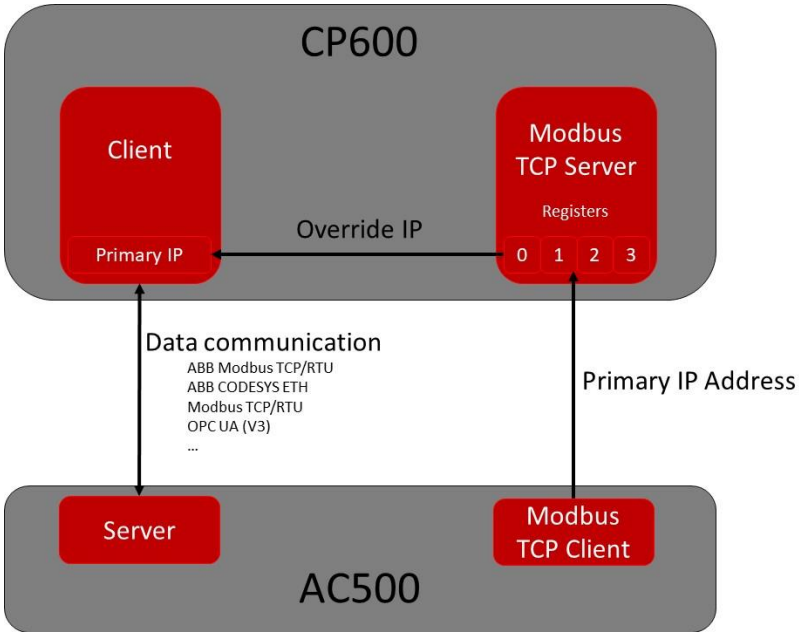
-When the connected PLC go offline/Stop

-When the connected PLC 'ask' to switch to secondary PLC

-Manual from a Panel button

The core of this functionality is executed by Generic Modbus protocol in which the HA CPUs act as Client. The CP600 device acts as Modbus TCP Server. Standard Modbus TCP messages are used for information when an exchange should be done.

This approach allows the CP600 to receive the Primary IP address of the HA system. Then the IP address is transferred to the special data type Node override IP which allows to change the IP address of the target controller (Primary CPU) at runtime.



Regarding the data communication between CP600 and HA CPU, we will choose the protocol' ABB Modbus TCP 'in the example project.

	Note: This Application example presumes that the user has already experiences and skills in conjunction with Automation Builder and Panel Builder, because detailed explanation of Automation Builder and Panel Builder basics are not part of this description and would go beyond the scope of this Application example.

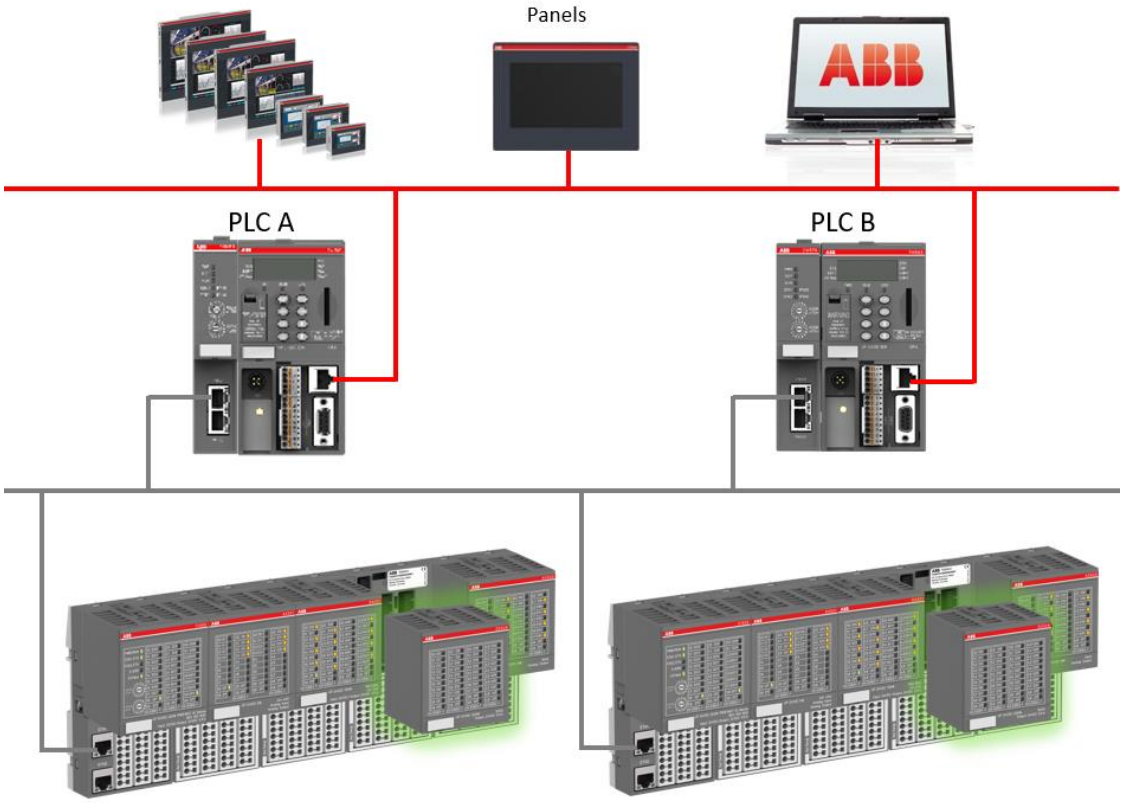
2.2 Compatibility

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.

- 2 identical AC500 V2 PLCs
- Automation Builder 2.4.0 or newer
- Panel Builder 600 V2.8.1 or newer

2.3 Overview

The figure below shows an overview of some involved components in our example project. Switches are not included in the network.



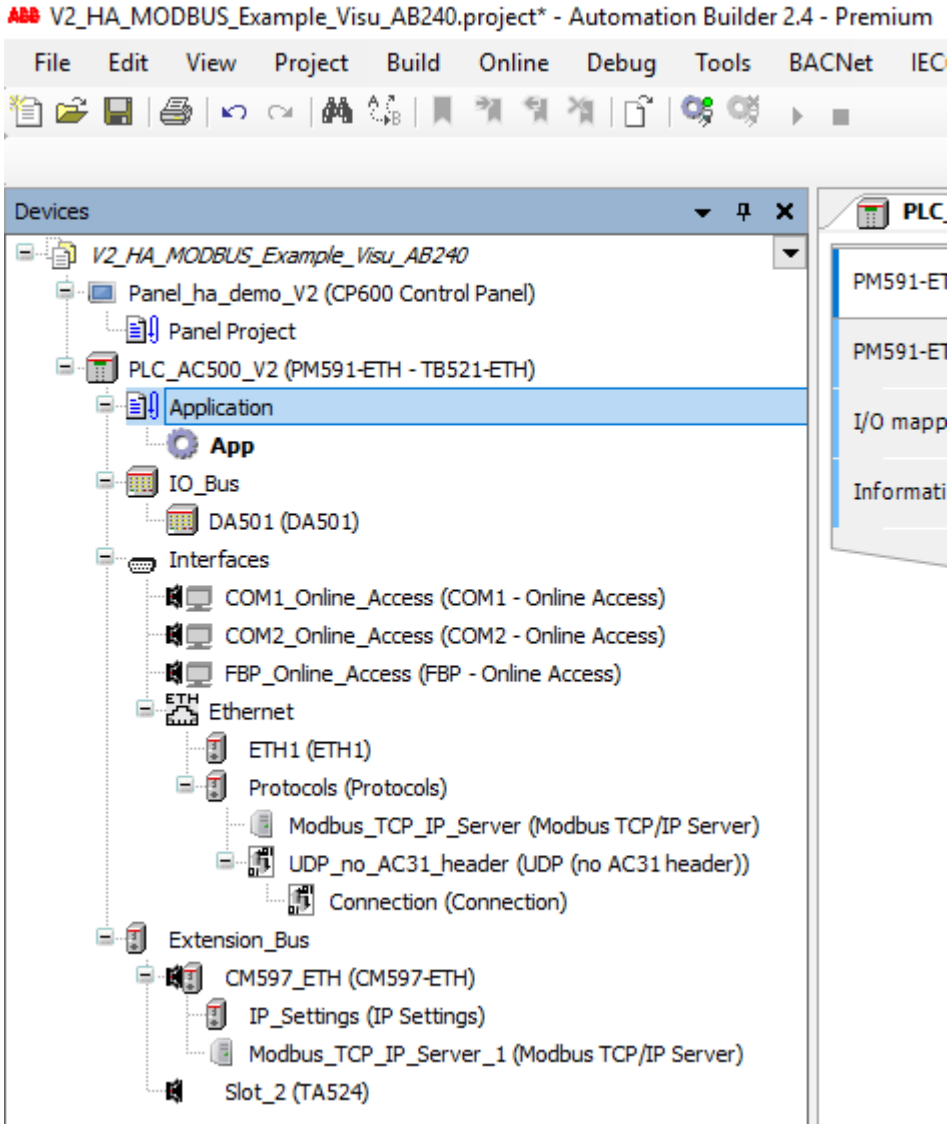
3 CP600 HA V2 configuration

The following description show the required steps for the AC500 as well for the Panel in detail.

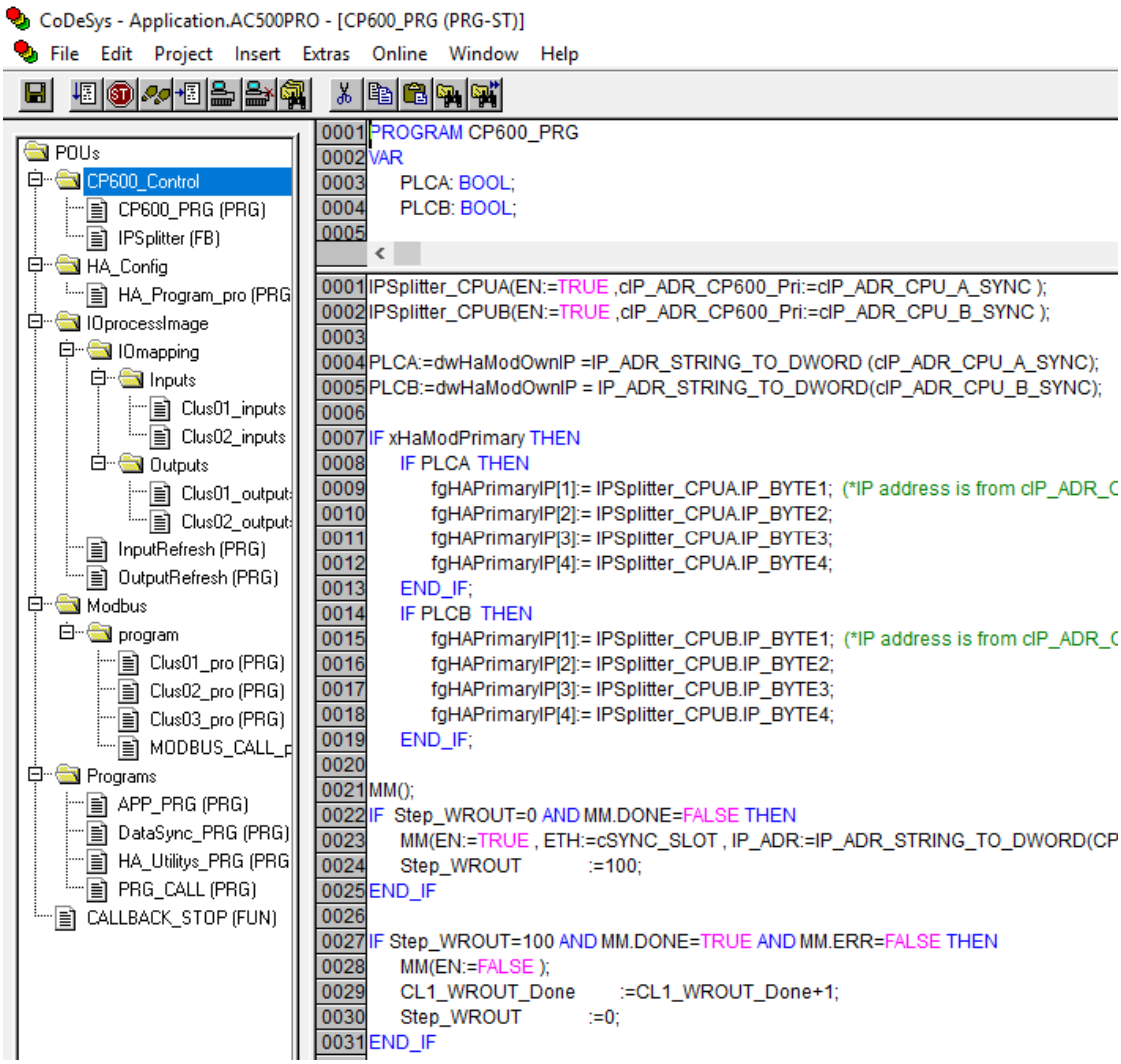
3.1 AC500 V2 HA example project

	Note: Programming software, user program, functions blocks and configuration data can be viewed and adapted / modified to your needs.

In the example ‘V2_HA_MODBUS_Example_Visu_AB240.project’, Double-click ‘Application’ to launch the CODESYS editor for the PLC as shown below:



Browse to the ‘POUs’ tab, add a new folder ‘CP600_Control’ and add a program and a function block within the folder. Name the new objects as shown below (e.g. CP600_PRG, FB IPSplitter):



Coding contents

IPSplitter(Function Block)

- Convert the IP address string to 4 Bytes output, please note the byte order in the CP600 device.

```
0001 FUNCTION_BLOCK IPSplitter
0002 VAR_INPUT
0003     EN :BOOL;
0004     clP_ADR_CP600_Pri:STRING[16]:= '192.168.2.10';
0005 END_VAR
0006 VAR_OUTPUT
0007     IP_BYTE1:BYTE;
0008     IP_BYTE2:BYTE;
0009     IP_BYTE3:BYTE;
0010     IP_BYTE4:BYTE;
0011     IP_BYTE_ARRAY:ARRAY[0..3] OF BYTE;
0012 END_VAR
0013 VAR
0014     clP_Dword: DWORD;
0015 END_VAR
```

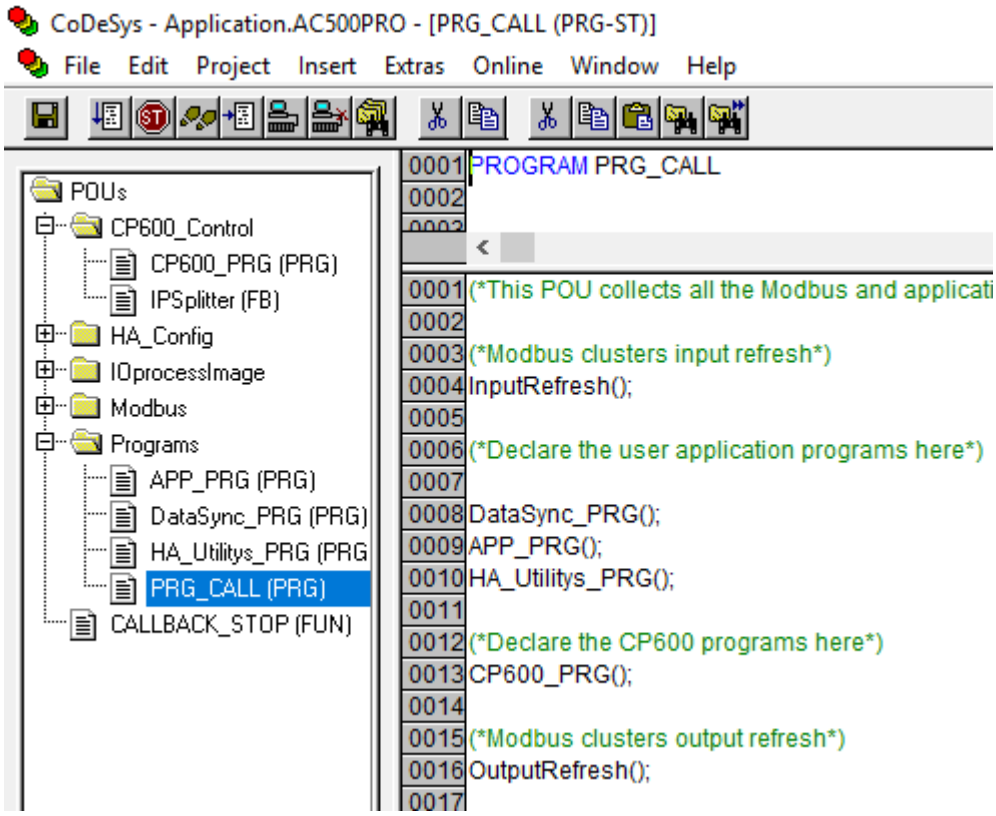
```
0001 IF EN THEN
0002     clP_Dword:=IP_ADR_STRING_TO_DWORD(clP_ADR_CP600_Pri);
0003     SysMemCpy(ADR(IP_BYTE_ARRAY), ADR(clP_Dword), 4);
0004     IP_BYTE1:=IP_BYTE_ARRAY[1];
0005     IP_BYTE2:=IP_BYTE_ARRAY[0];
0006     IP_BYTE3:=IP_BYTE_ARRAY[3];
0007     IP_BYTE4:=IP_BYTE_ARRAY[2];
0008 END_IF
```

CP600_PRG(PRG)

- Reads out the own IP address of CPU.
- Write the own IP address with 4 bytes format to the corresponding registers in the CP600 device (e.g. registers 0000,0001,0002,0003), if CPU is in Primary.

```
0022 IF  Step_WROUT=0 AND MM.DONE=FALSE THEN
0023     MM(EN:=TRUE , ETH:=cSYNC_SLOT , IP_ADR:=IP_ADR_STRING_TO_DWORD(CP600_IP_ADR) ,UNIT_ID:=0 , FCT:=16 , ADDR:=16#0 , NB:=2 , DATA:=ADR(fgHAPrimaryIP));
0024     Step_WROUT          :=100;
0025 END_IF
```

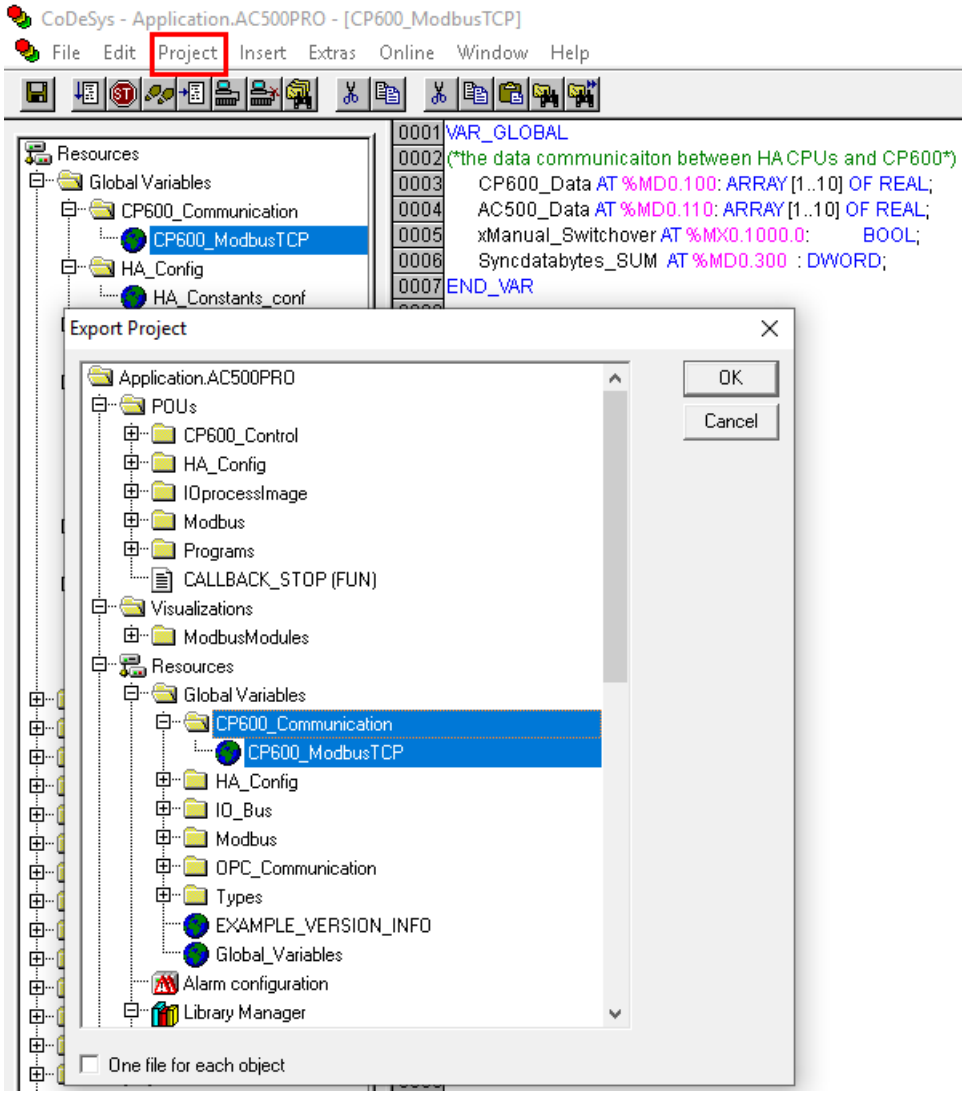
After that, append the program call in PRG_CALL and close the window.



	Note: Users can create a new AC500 V2 HA project instead of opening the attached example project, then add the above coding accordingly. For details on how to program the AC500 V2 HA, please refer to the document and example in the folder C:\Users\Public\Documents\AutomationBuilder\Examples\PS5601-HA-MTCP

3.2 Exporting tags from the controller

Automation Builder programming supports tag export in xxx.exp format.
Select ‘Project -> Export… ‘from CODESYS editor to export the global variables ‘CP600_ModbusTCP’ with named ‘CP600.exp’.

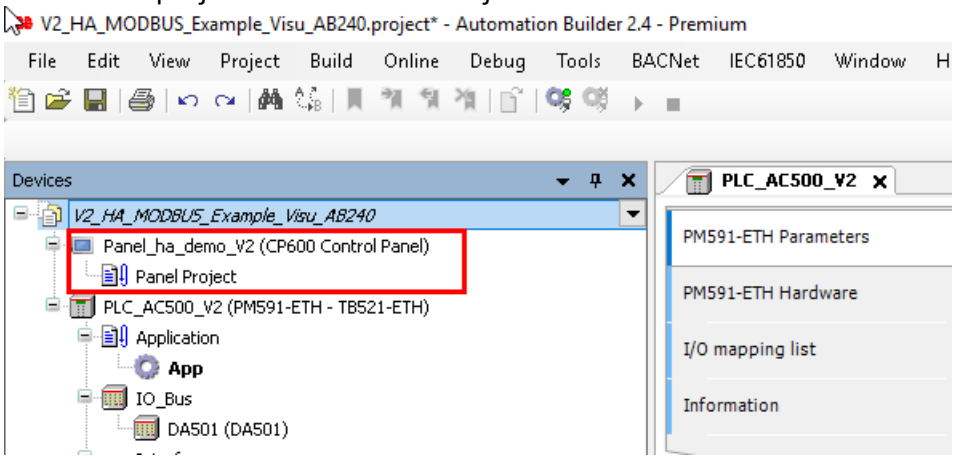


3.3 CP600 HA example project

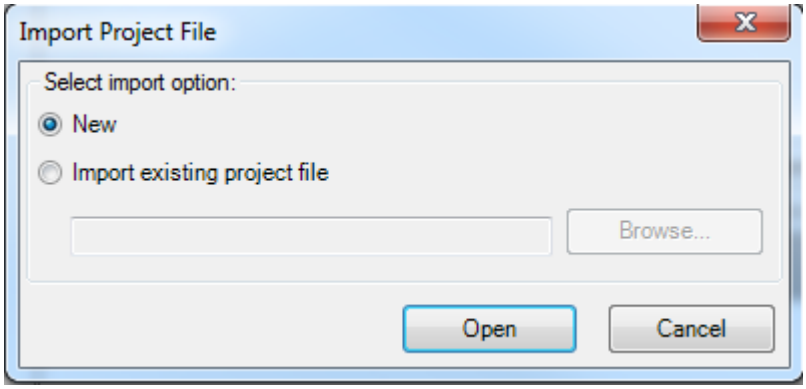
In this chapter, we will add the CP600 panel project into the example project.

3.3.1 Create CP600 Panel project

Right-click 'V2_HA_MODBUS_Example_Visu_AB240' from the Devices tree in the Automation Builder project and select 'Add object' to choose the CP600 Control Panel.



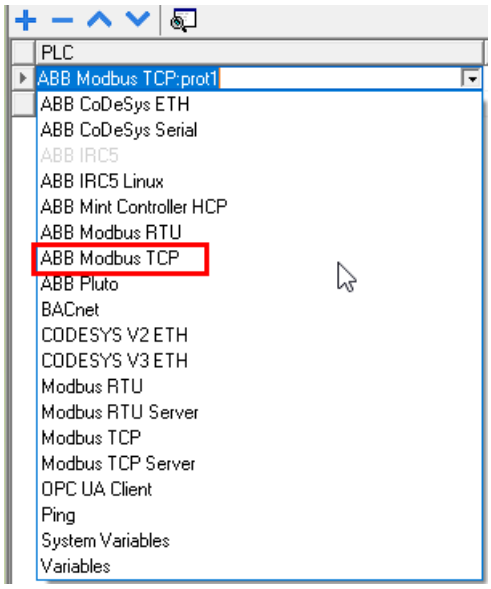
Double-click ‘Panel Project’. Uncheck the ‘‘Connect’ box and uncheck the ’Update Panel Builder project on launch’. This is because we don’t use the ABB CODESYS V2 ETH protocol.
Click the ‘Launch Panel Builder Editor’
In the Pop-up Windows, select ‘New’ and click ‘Open’ to continue.



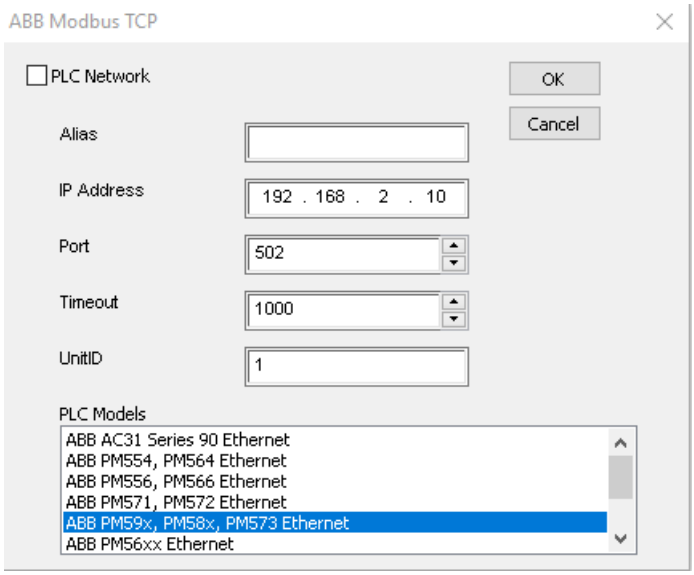
At the project wizard, select the CP6610 panel and click ‘Finish’ to continue.

3.3.2 Communication protocol ‘ABB Modbus TCP’ and Tags.

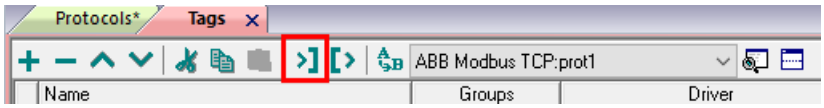
Double-click “Protocols” to open the editor.
Click on the “+” to add the protocol and select “ABB Modbus TCP” in the dropdown list.



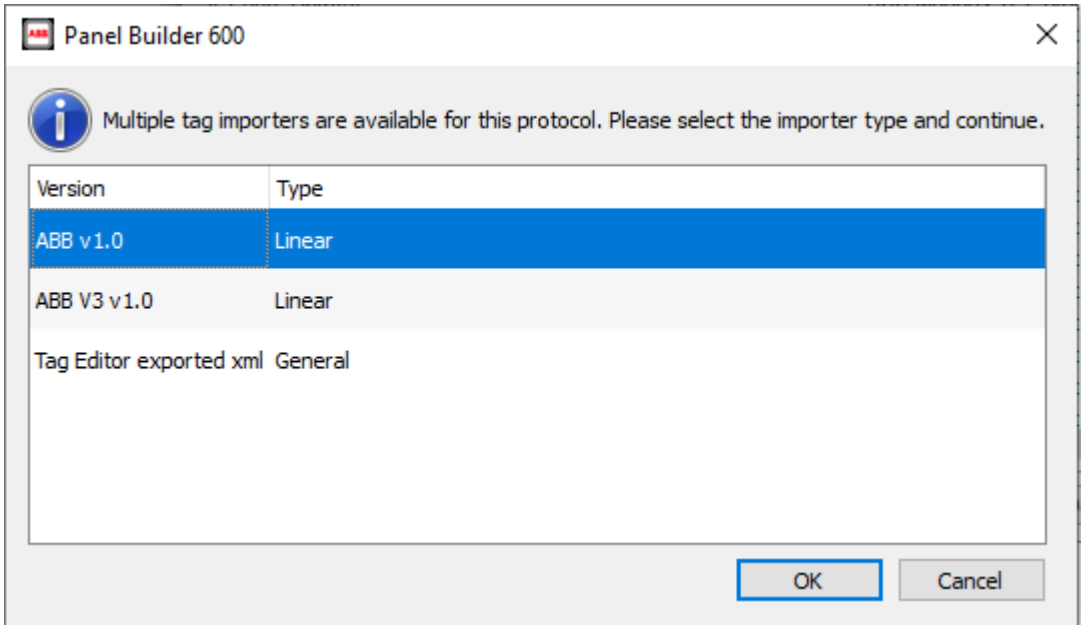
At the ABB Modbus TCP window, type in the ‘IP address’ with any desired IP address (e.g. the IP address could be one of the CPUs).



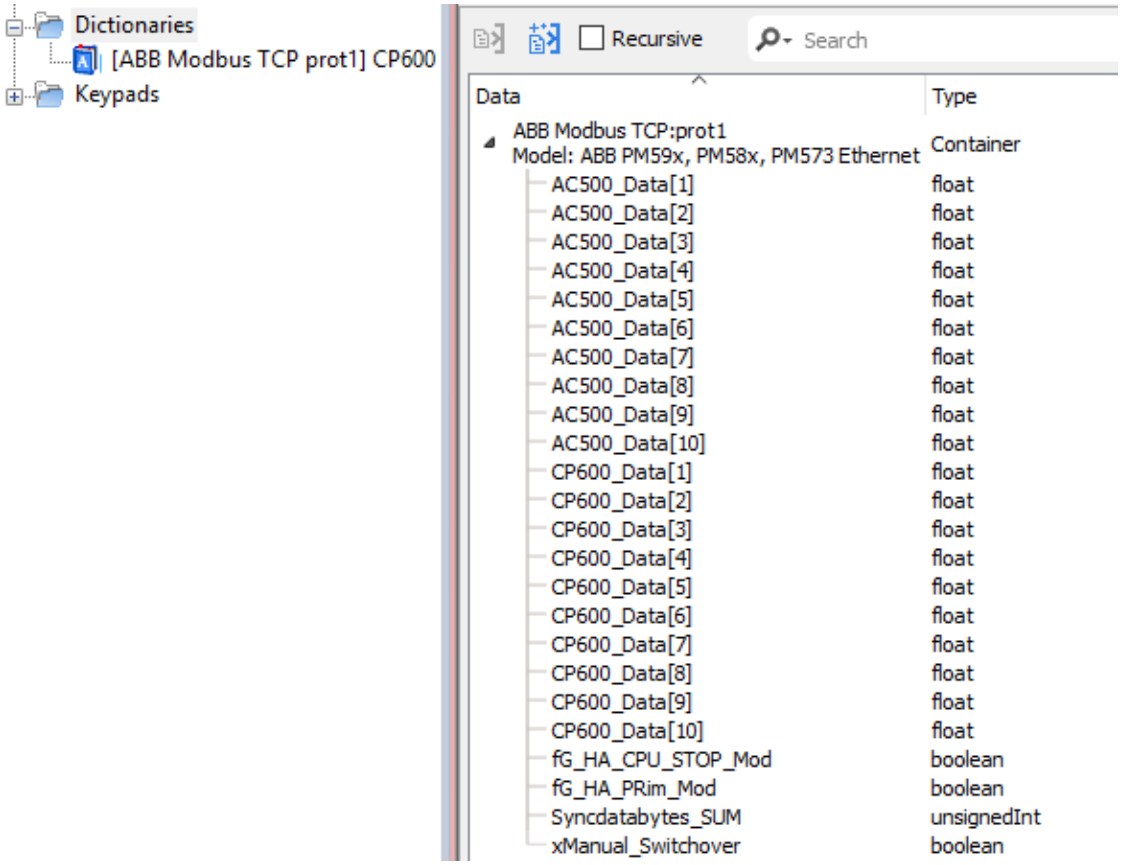
Click “Ok” button to continue and save the project.
Double click on the “Tags” to open the editor, then click on the “Import tags” button.



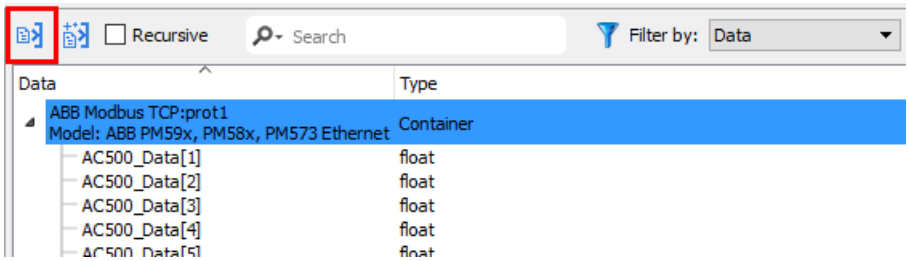
Select the ABB v1.0 and click “Ok” button to continue.



Type or browse to the file ‘CP600.exp’ exported from CODESYS Editor and click Open to continue.
The tags now appear in the Dictionaries.

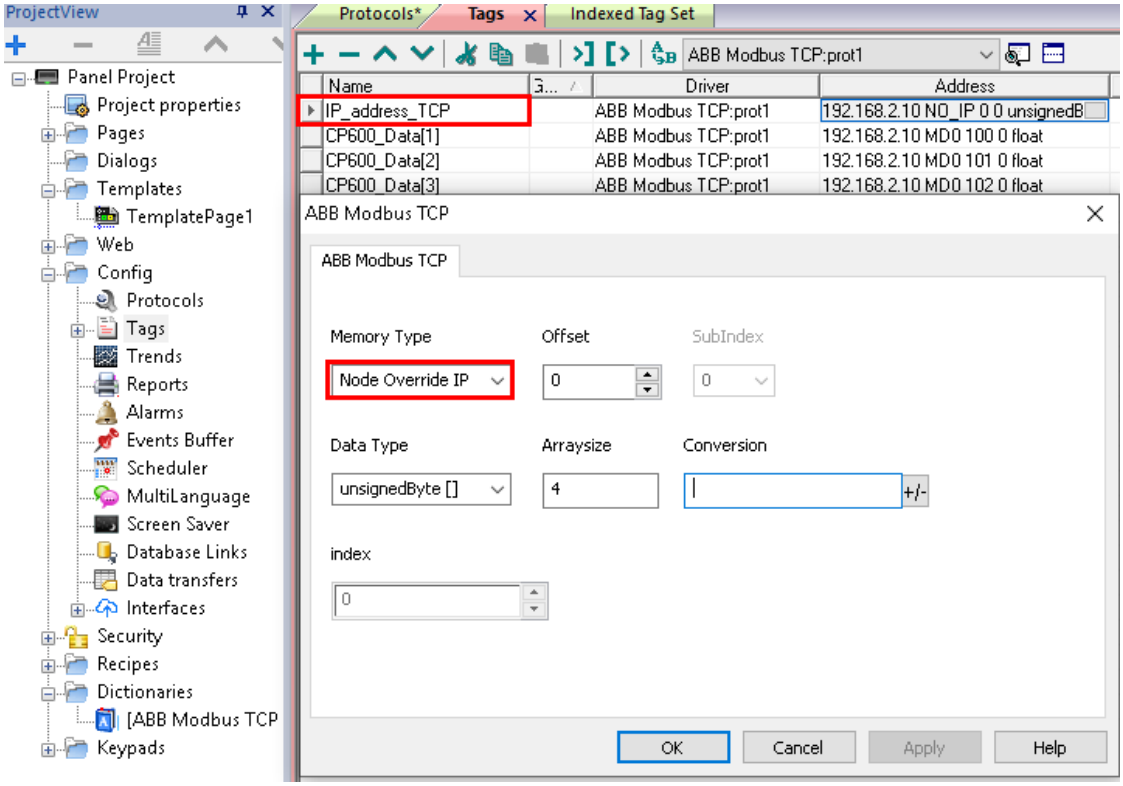


Select the tags you want and click on the “Import Tags” button.



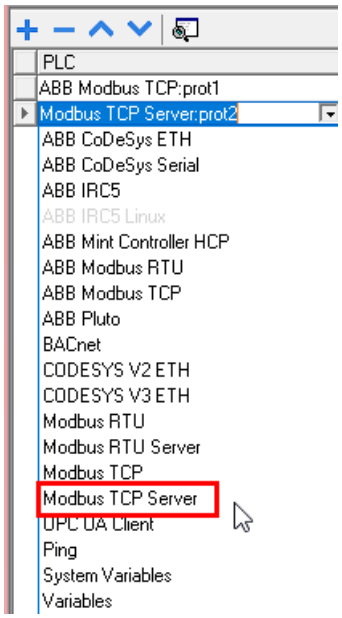
During system operation the Primary CPU might change. The Panel should always communicate with the Primary CPU. To archive this, we add one tag with type Node Override IP, in the example ‘IP_address_TCP’, which allows you to change the IP address for the target CPU in the panel during runtime.

The data type is an array of 4 unsigned bytes, one per each byte of the IP address.

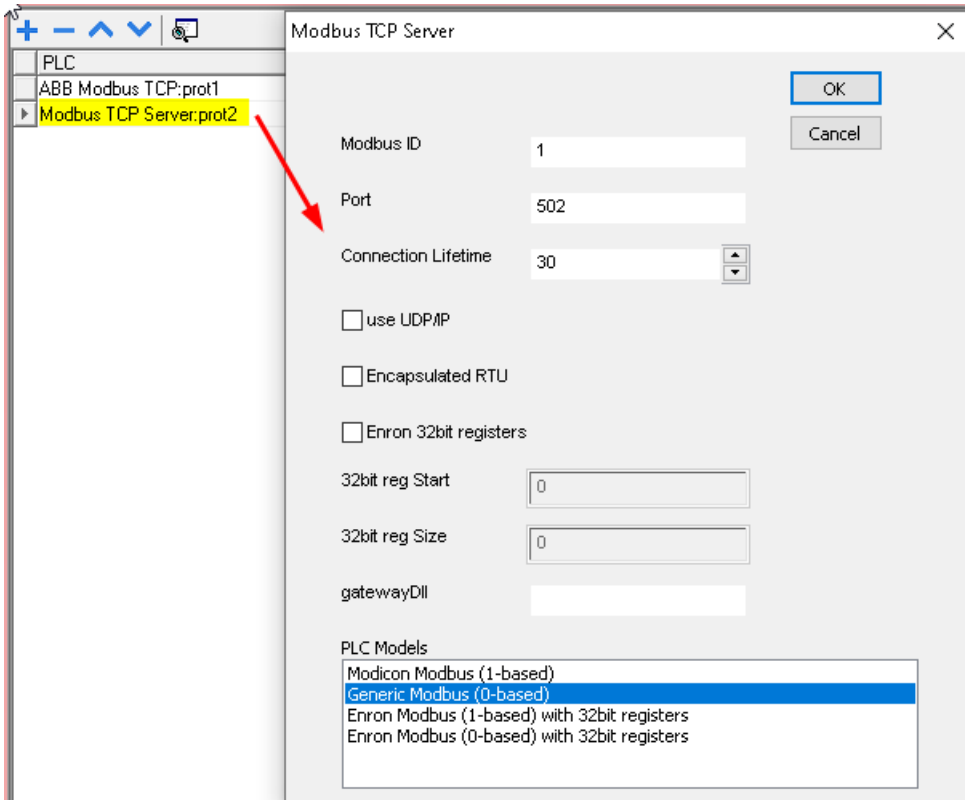


3.3.3 Communication protocol ‘Modbus TCP Server’ and Tags.

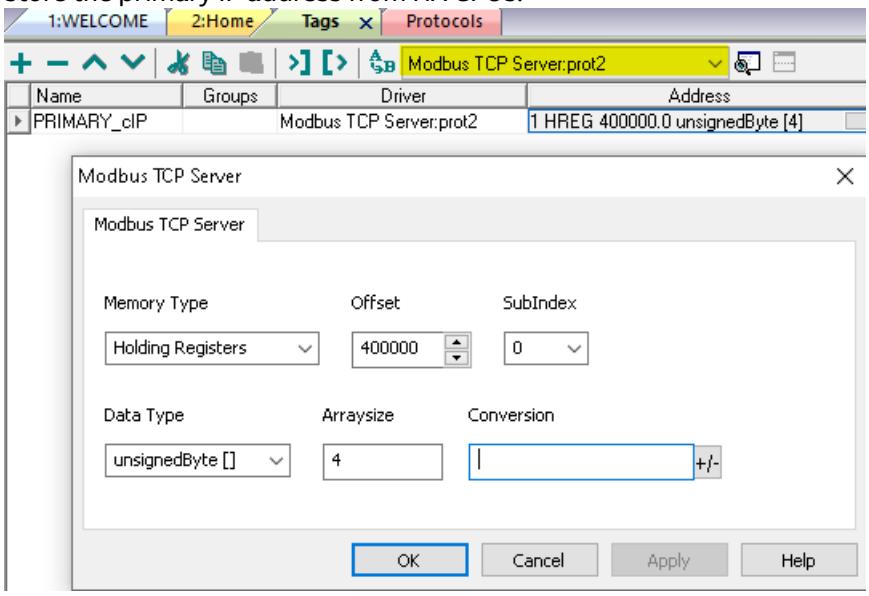
This communication driver implements a Modbus TCP Server unit in CP600 device. A subset of the complete range of Modbus function codes is supported. The available function codes allow data transfer between clients on the TCP network and the server. The CP600 device acts as a server in the network.
Double-click on the “Protocols” to open the editor again.
Click on the “+” to add the option “ModbusTCP Server” in the dropdown list.



At the Modbus TCP Server window, select Generic Modbus(0-based) for PLC models.

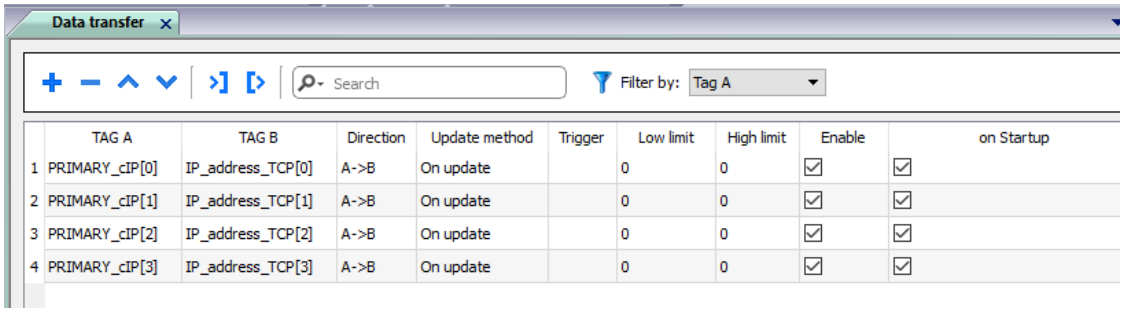


Create one tag named PRIMARY_cIP for the project as below. This array tag will receive and store the primary IP address from HA CPUs.



3.3.4 Data transfers

Data transfer allows you transferring variable data from one device to another. Using this feature an CP600 device can operate as a gateway between two devices, even if they don't use the same communication protocol.
Double-click 'Data transfers' to use the editor to map transfer rules.
In TAG A column we add the Tag 'PRIMARY_cIP' that was created in protocol Modbus TCP Server.
In TAG B column one tag 'IP_address_TCP' with type Node Override IP will be added, which allows you to change the IP address for the target CPU in the panel during runtime.

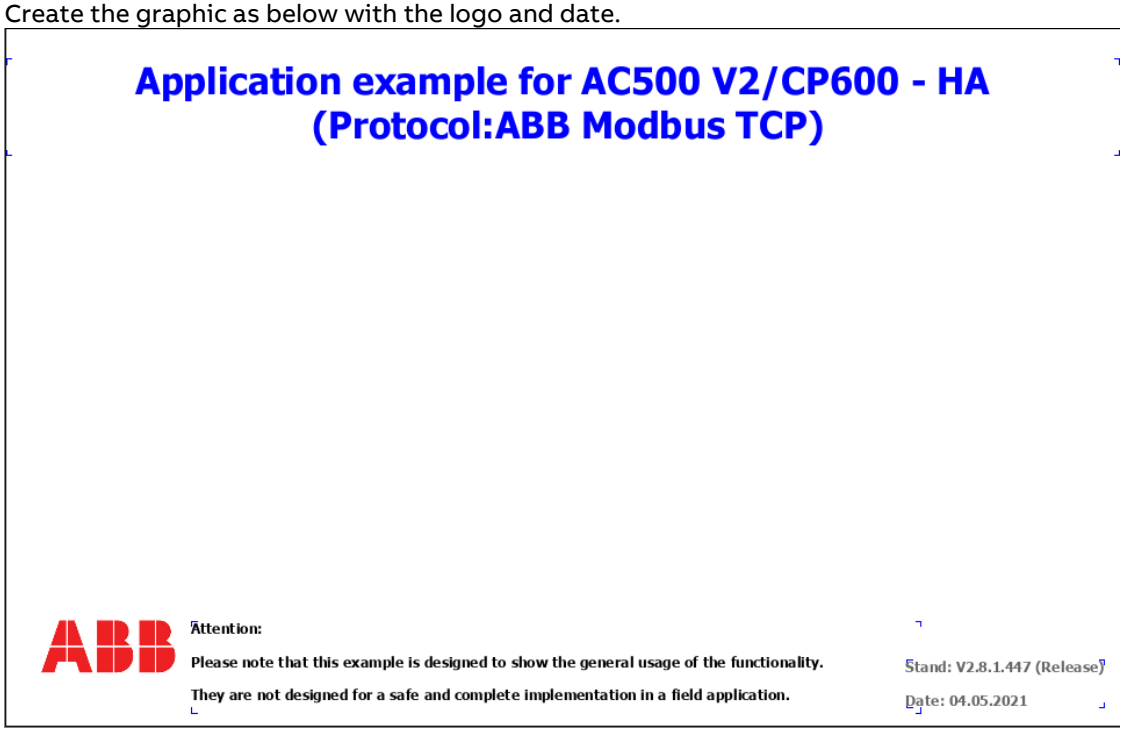


	TAG A	TAG B	Direction	Update method	Trigger	Low limit	High limit	Enable	on Startup
1	PRIMARY_cIP[0]	IP_address_TCP[0]	A->B	On update		0	0	☑	☑
2	PRIMARY_cIP[1]	IP_address_TCP[1]	A->B	On update		0	0	☑	☑
3	PRIMARY_cIP[2]	IP_address_TCP[2]	A->B	On update		0	0	☑	☑
4	PRIMARY_cIP[3]	IP_address_TCP[3]	A->B	On update		0	0	☑	☑

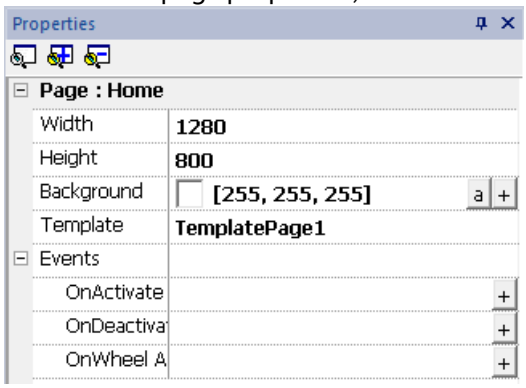
The ‘On update’ method allows changing the values in accordance with the direction settings only when the source value changes. The default value of the update rate of each tag is 500ms and can be modified with Tag editor.

3.3.5 Pages Setup

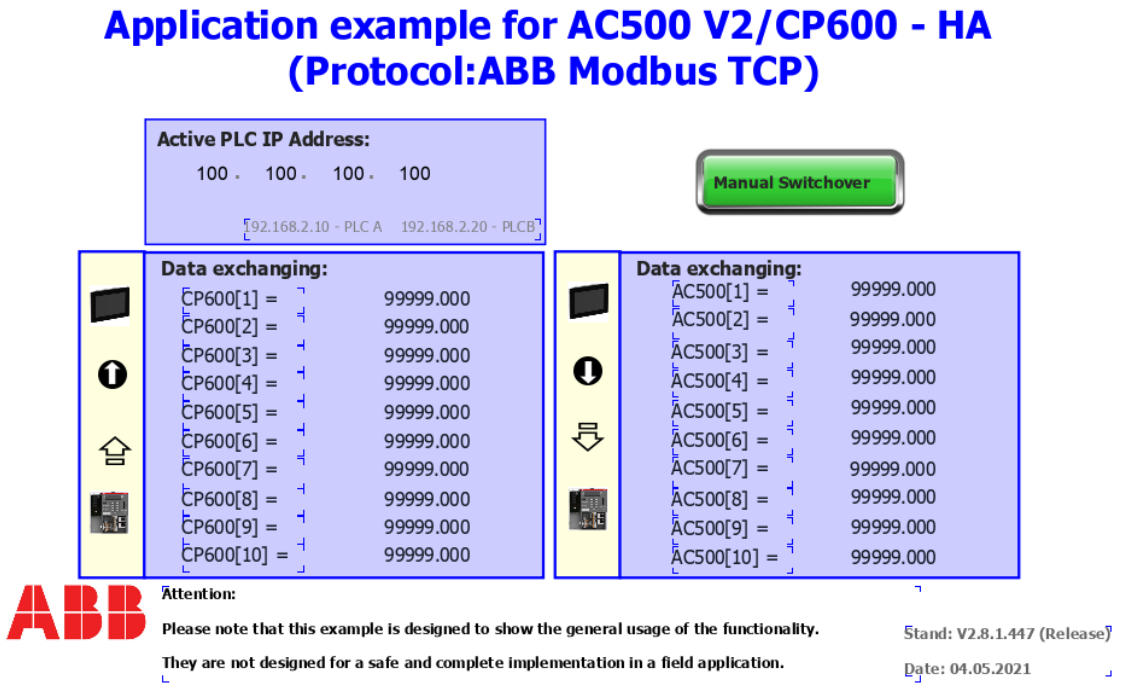
Right-click on the ‘Templates ’folder and select ‘insert New Template page’.
At the “New Page”, leave the name as default and click “Ok” button to continue.
Create the graphic as below with the logo and date.



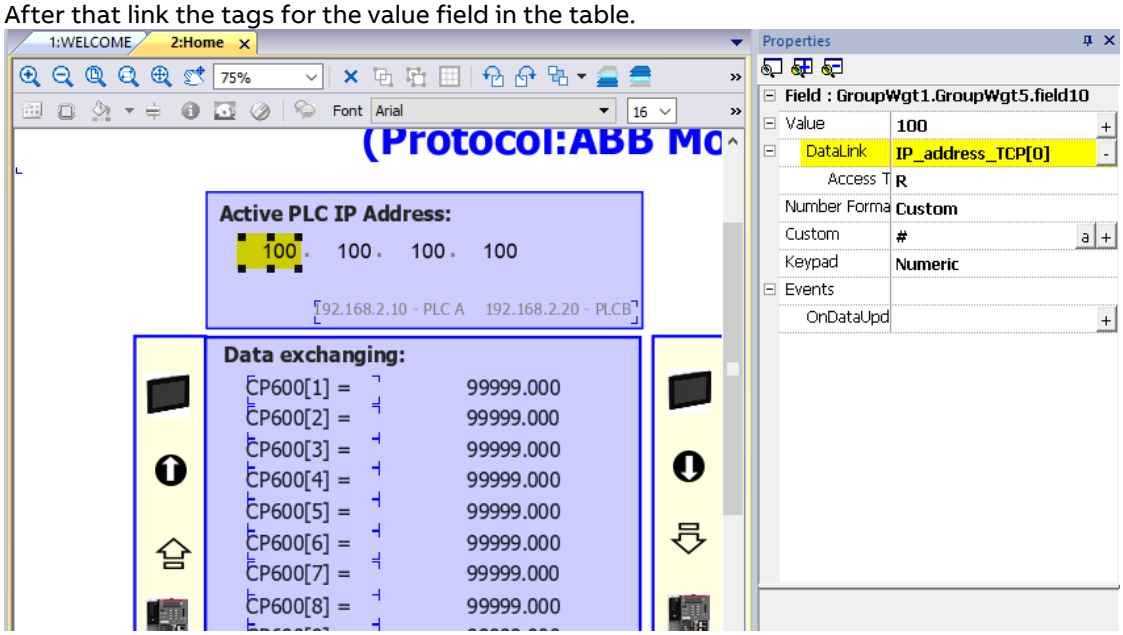
Now we will create the Home Page
At the Home page properties, select the “Templatepage1” from the drop down menu.



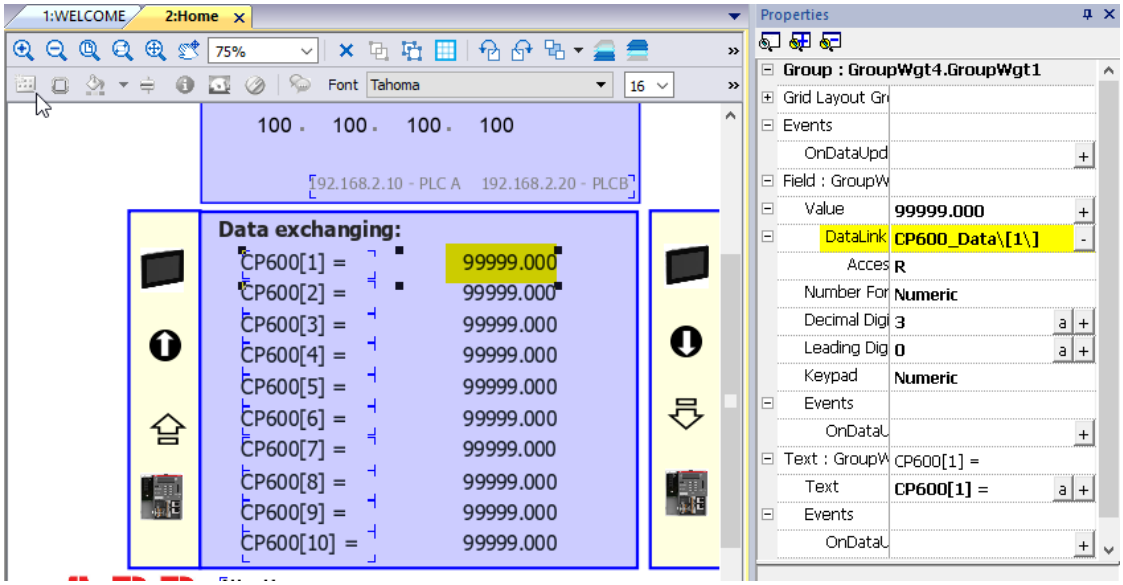
Draw the graphic as below.



After that link the tags for the value field in the table.



Repeat the same for others.
The “Active PLC IP Address” box is showing the IP Address of the connected PLC - Primary CPU.



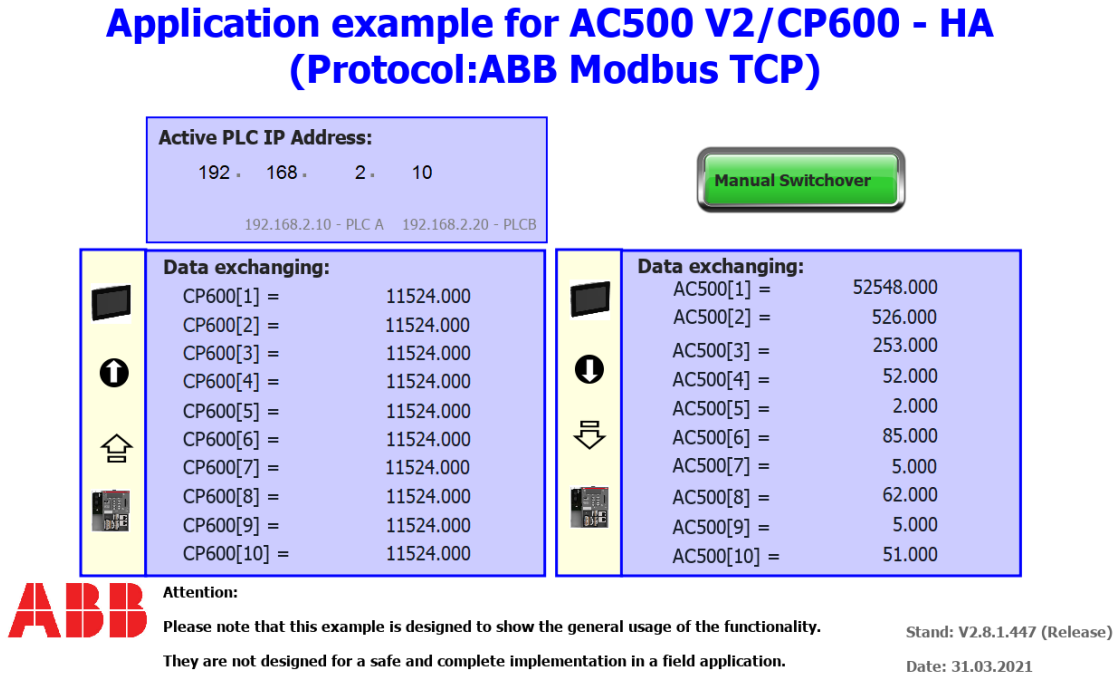
You can find here few test variables and the values of the PLC tag used to force the PLC switching from the connected PLC.

To make test, you can disconnect the PLC from network or change values inside the PLC memory to verify the both PLCs Swapping function.

When complete, download the project to the CP600 panel.

3.4 Application Example test program

The application example project contains the complete configuration of HA and CP600 Panel. You can see the data exchange between CP600 Panel and HA Primary CPU as shown below.



4 Appendix

The CP600 will lose communication if the ethernet cable connected to the Primary CPU is broken/disconnected. The HA CPUs don't switch over in this situation automatically. The CPU with a broken ethernet cable is still active/primary status. The workaround is to use the ping function block in the application to detect the CP600 device so that manual changeover from primary to secondary PLC.

The codes of Ping FB please refer to the below picture that will not be mentioned in example project.

```
0001 (*   This step chain controls via ICMP (Ping), whether the IP adress of the Modbus-TCP servers is valid.*)
0002
0003 IF step = 0 THEN
0004     step    := 10;
0005 END_IF
0006
0007 IF step = 10 AND NOT ETH_ICMP_PING_1.DONE  THEN                    (*Wait for ETH_ICMP_PING done*)
0008         MOD_Ping_IP_ADR   := '192.168.2.100';
0009         MOD_Ping_EN       := TRUE;
0010         step              := 20;
0011 END_IF
0012
0013 ETH_ICMP_PING_1(EN:= MOD_Ping_EN,ETH:=11,IP_ADR:=IP_ADR_STRING_TO_DWORD(MOD_Ping_IP_ADR) ,TIMEOUT:=MOD_Ping_TIMEOUT);
0014
0015 IF ETH_ICMP_PING_1.DONE THEN
0016     IF ETH_ICMP_PING_1.ERR THEN
0017         MOD_Ping_ERNO         := ETH_ICMP_PING_1.ERNO;
0018         MOD_Ping_ERR_Count    := MOD_Ping_ERR_Count +1;
0019     ELSE
0020         MOD_Ping_OK_Count     := MOD_Ping_OK_Count +1;
0021     END_IF
0022 END_IF
0023
0024 IF step = 20 AND  ETH_ICMP_PING_1.DONE THEN
0025     IF ETH_ICMP_PING_1.ERR THEN
0026         Slot11_Ping_IP_OK   := FALSE;
0027     ELSE
0028         Slot11_Ping_IP_OK   := TRUE;
0029     END_IF
0030     MOD_Ping_EN     := FALSE;
0031     step            := 10;
0032 END_IF
```

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