Part I

Training 1

1 AC500

The following electronic modules are the basic components of the AC500 system:

- CPUs of different performance classes
- Communication Modules for various bus systems, e.g. PROFIBUS®, CANopen®, DeviceNet™, PROFINET®, EtherCAT®

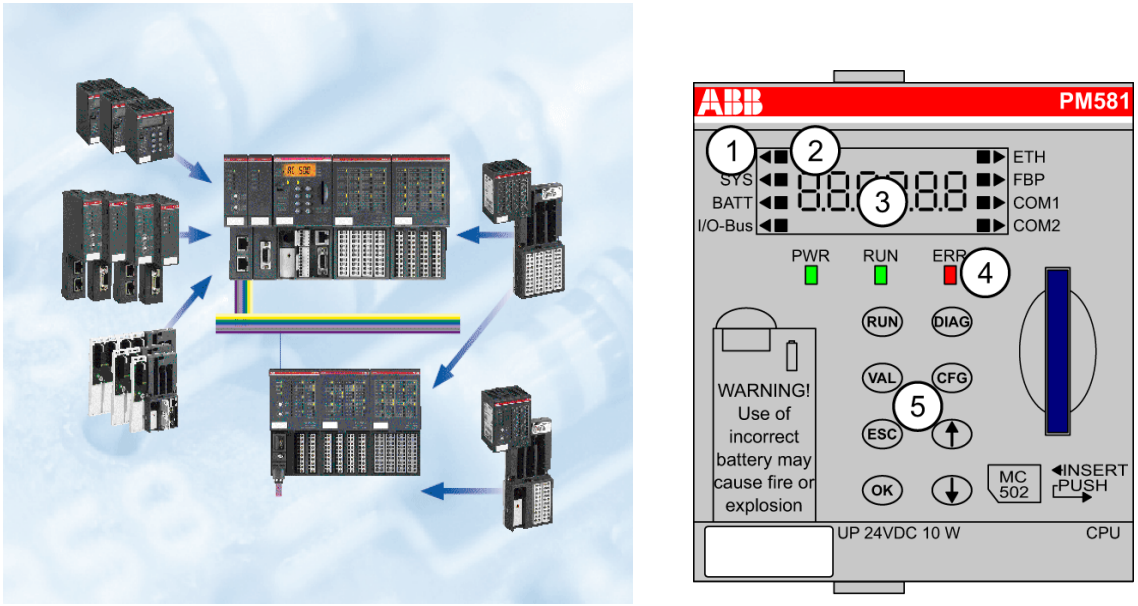

Figure 1: PLC Systems

Central expansion: One CPU supports direct connection of up to seven S500 FBP system I/O devices.

Decentralized expansion: Using the DC505 FBP interface module (for fieldbus systems PROFIBUS® DP, CANopen®, DeviceNet™, Modbus RTU), up to seven S500 FBP system I/O devices can be connected (with a maximum of four analog modules).

CPU PM573 (512kB, 2 RS-232/485 interfaces - programming, Modbus/CS31 (COM1), 1 x FBP) with an integrated Ethernet Communication Module (TCP/IP).

The I/O Modules Digital and analog in different versions. Can be simply plugged onto the terminal units – for local expansion of the CPU (max. 10 modules).

Communication Modules Communication Module for fieldbus systems - PROFIBUS® DP, DeviceNet™ and CANopen®.

AC500 Control Builder provides the following functionalities:

- Five standardized programming languages: Function Block Diagram (FBD), Instruction List (IL), Ladder Diagram (LD), Structured Text (ST), Sequential Function Chart (SFC);
- Free graphical function chart (CFC);
- Debugging functions for the program test: Single step, Single cycle, Breakpoint.

IEC 61131-3 commands can be simulated without a PLC being connected, including the relevant malfunctions. After the program test, the application can be downloaded to the control system.

2 Introduction to the program environment

1 Declaration of the Modules

Start ABB Automation Builder.

This is an environment to connect ABB controller with CoDeSys Programming System. Then program starts you will see recent projects. If you want to start your own project - you should choose the right configuration of the controller.

Choose the name of your project and location where it will be saved, see Fig. 2.

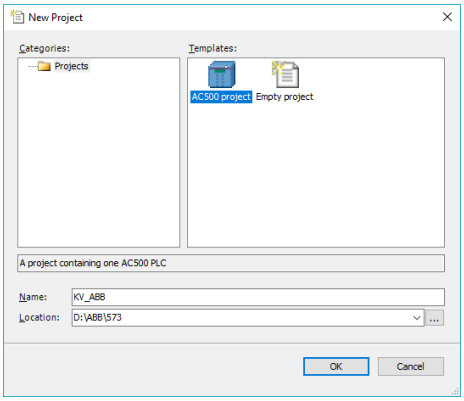

Figure 2: Start New Project

Look at the main module top right coner where type of the PLC is written. Select **AC500 PM573-ETH, AC500 PM590-ETH**, or **AC500 PM554-ETH** see Fig. 3.

To avoid some error messages

Battery Click on PLC_AC500_V2 and open the PM5XX-ETH Parameters window:

Set new parameter: Check battery--value = **OFF**.

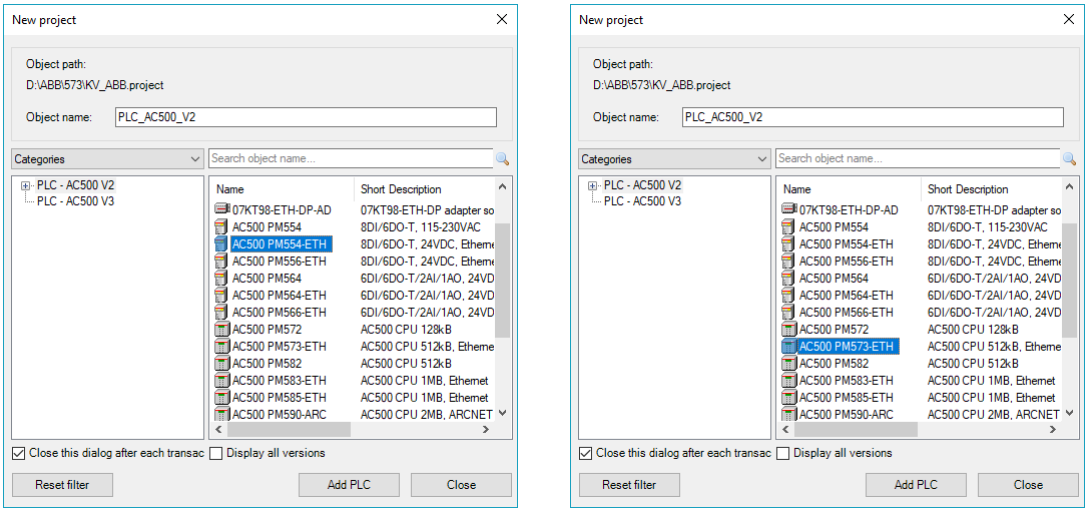

Figure 3: Controller's types

1.1 I/O

Depending on the type of the controller Map the inputs and outputs. As it is done physical I/Os will be available in your project as Global Variables.

PM573/590-ETH

In the Devices Window declare the right I/O module

- Click with right button on IO_Bus and choose Add Object.

- In the new window in the tree S500-I/O modules choose DA501.

Double click on the DA501 opens the window DA501 Parameters.

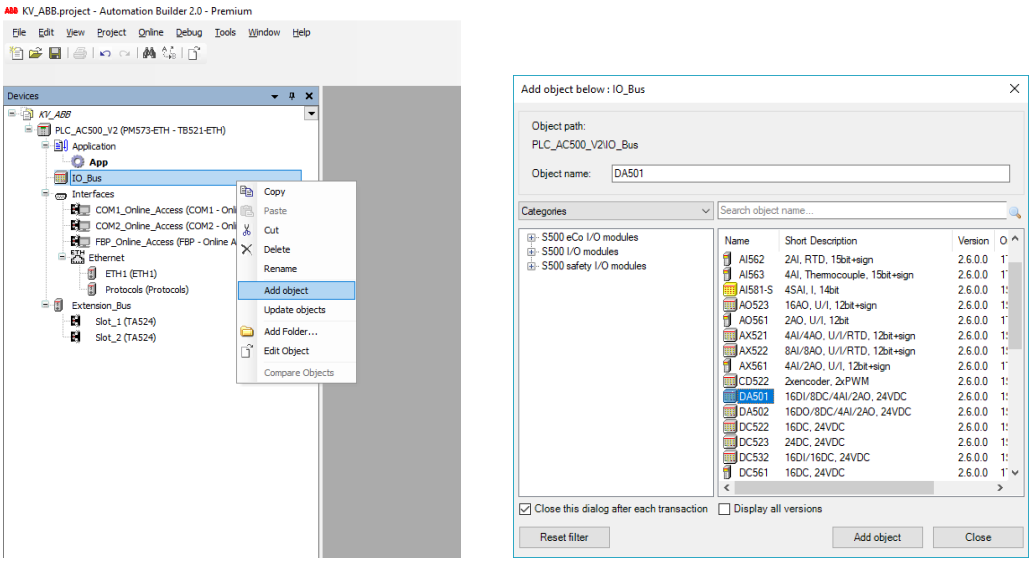

Figure 4: Adding the new Module

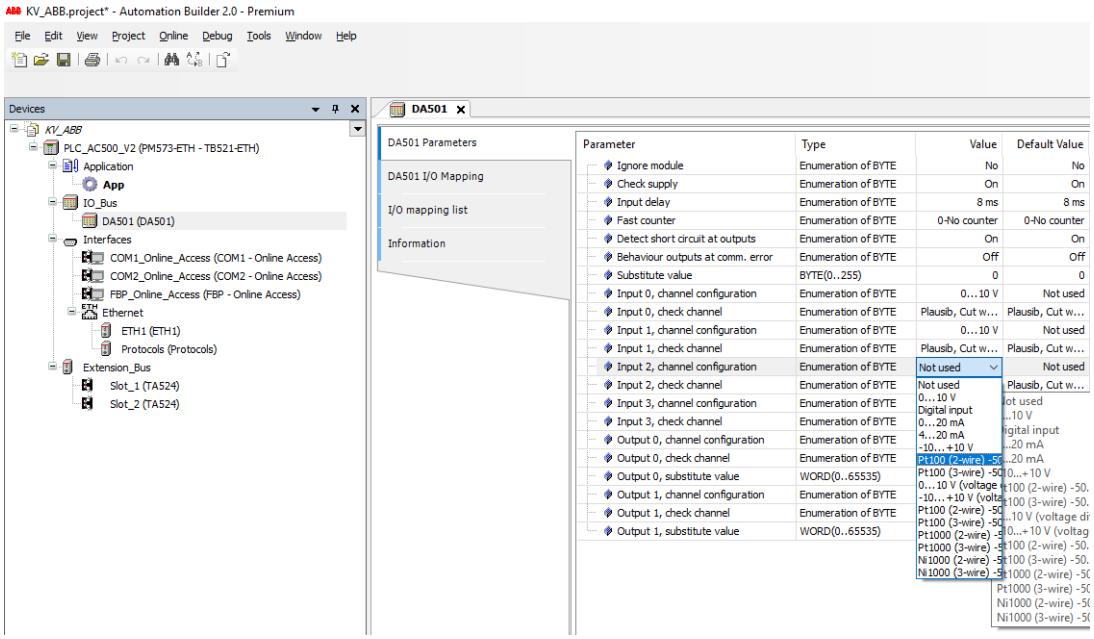

Figure 5: Analog I/O characteristics

First of all we need to set the right analog I/O parameters (Fig. 5):

- Input 0, channel configuration--value="0...10 V"
- Input 1, channel configuration--value="0...10 V"
- Input 2, channel configuration--value="Pt100(2-wire/ $-50\ldots+70/°C$)"
- Output 0, channel configuration--value="$-10\ldots+10/V$"
- Output 1, channel configuration--value="$-10\ldots+10/V$"

PM554-ETH

Controller PM554 already has some built in inputs and outputs. Thus, where is no any additional I/O module installed.

Mapping

On order to use Global Variable Names as names of module physical outputs, mapping is needed. Open the DA501 I/O Mapping or 8DI+6DO I/O Mapping window.

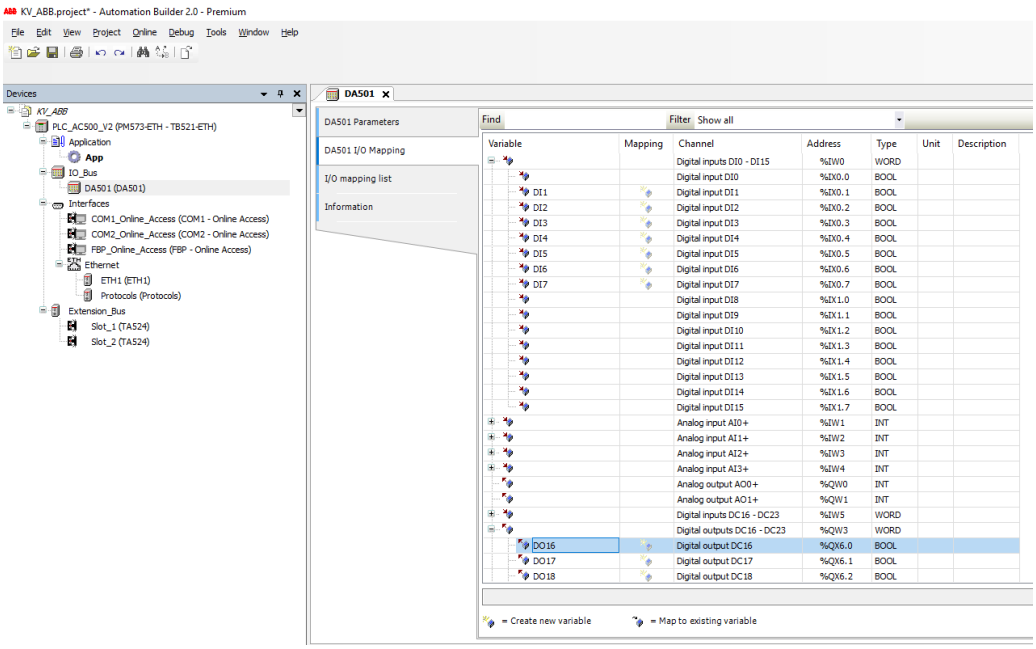

Figure 6: DA 501 I/O Mapping

- As first digital inputs \%IX0.0--%IX0.7) are connected to the switches S0-S7 / D0-D7, create the same variable names.

  - Digital inputs %IX1.0--%IX1.7, name variables according to the rules of the company. For example: m2_DI_8, where "m2" is the module position on the BUS, "DI"--Digital Input and "8"--position of the input.

- Analog Inputs \%IW1--\%IW4.

- Analog Outputs \%QW0--\%QW1.

- Digital Inputs \%IX10.0--\%IX10.7.

- Digital Outputs \%QX6.0--\%QX6.7 name as DO16–DO20.

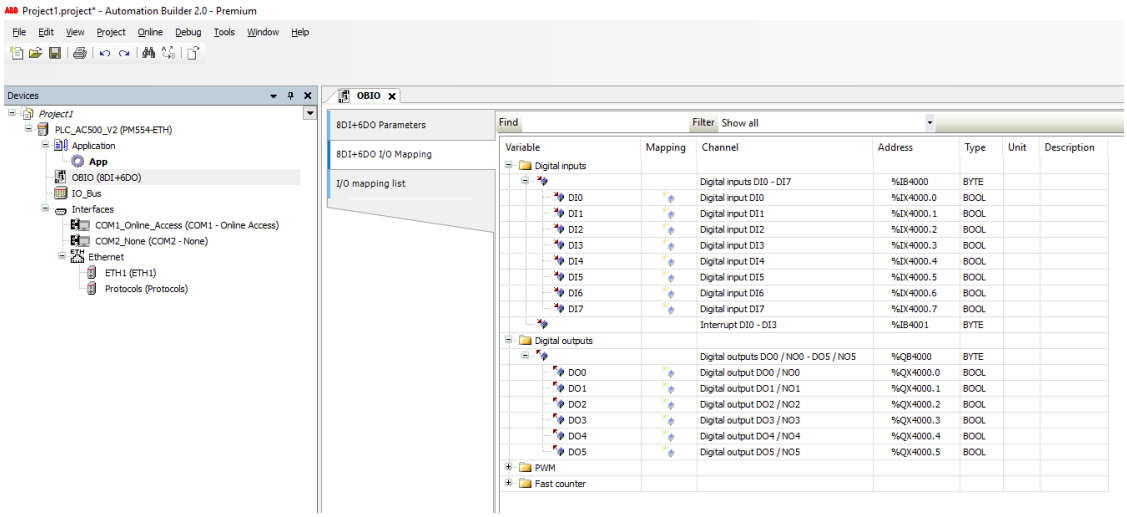

Figure 7: OBIO Mapping

2 Communication

To download your programs to the controller you need set the connection.

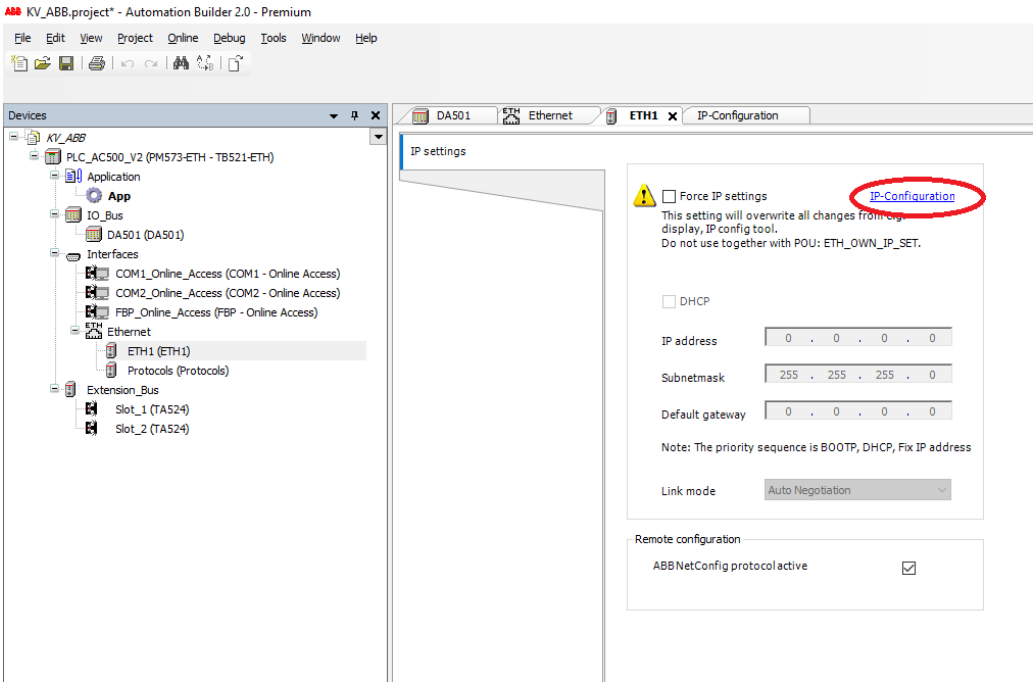

Figure 8: Ethernet

Open IP Configuration window by clicking ETH1 parameter in the Project tree (Fig. 8).

Scan available devices by IP address. Make sure, that IP address which is printed on PLC device is visible after scan, see Fig. 9.

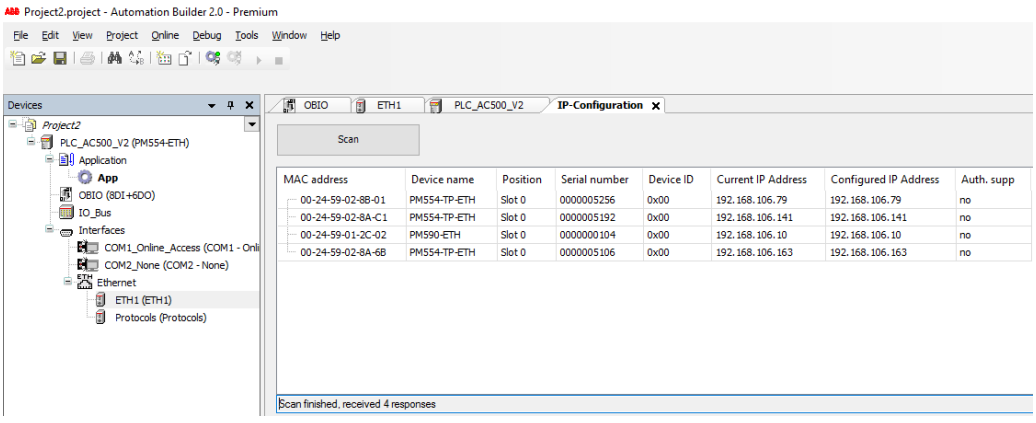

Figure 9: IP scan

Connect your project with a controller. Right Click on PLC_AC500_V2 and choose Commu-

nication Settings (Fig. 10).

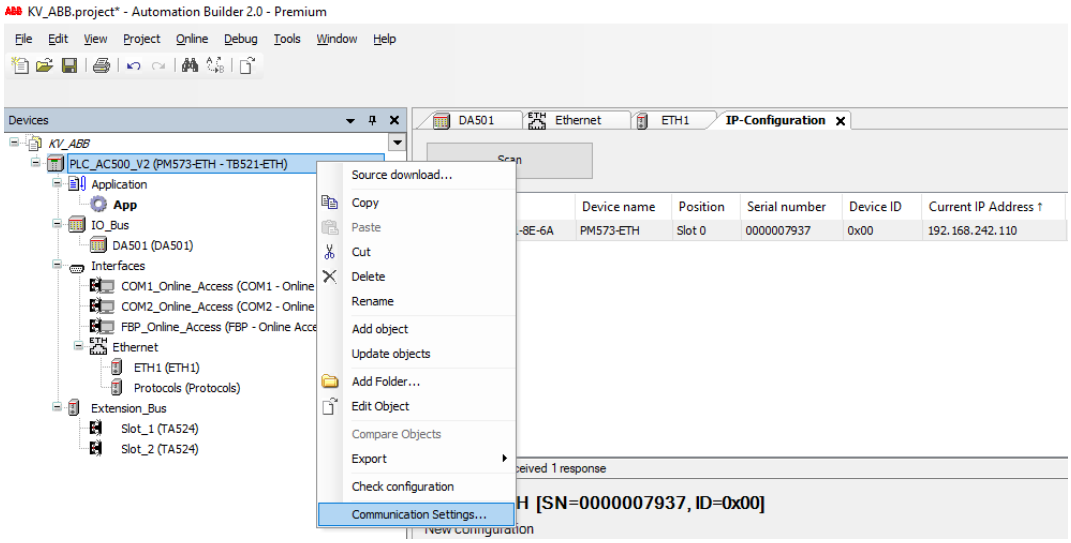

Figure 10: Communication Settings

You just need to add the IP address of your controller to the open window. If you would like to know more about the connection settings, then you need to check Use advanced settings., see Fig. 11.

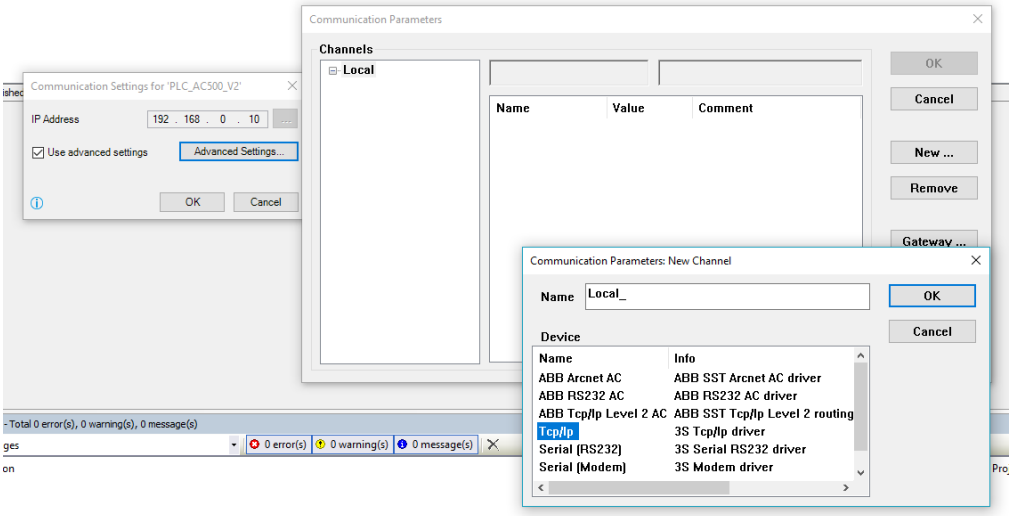

Figure 11: Advanced Communication Settings

List of the parameters can be seen in Fig. 12.

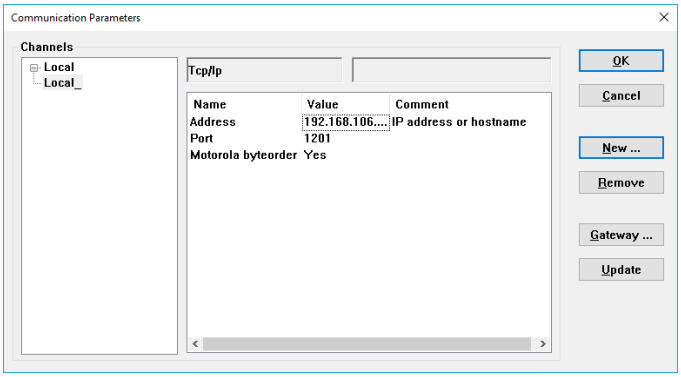

Figure 12: Communication Parameters

In order to check if connection is well established login to the controller, see Fig. 13.

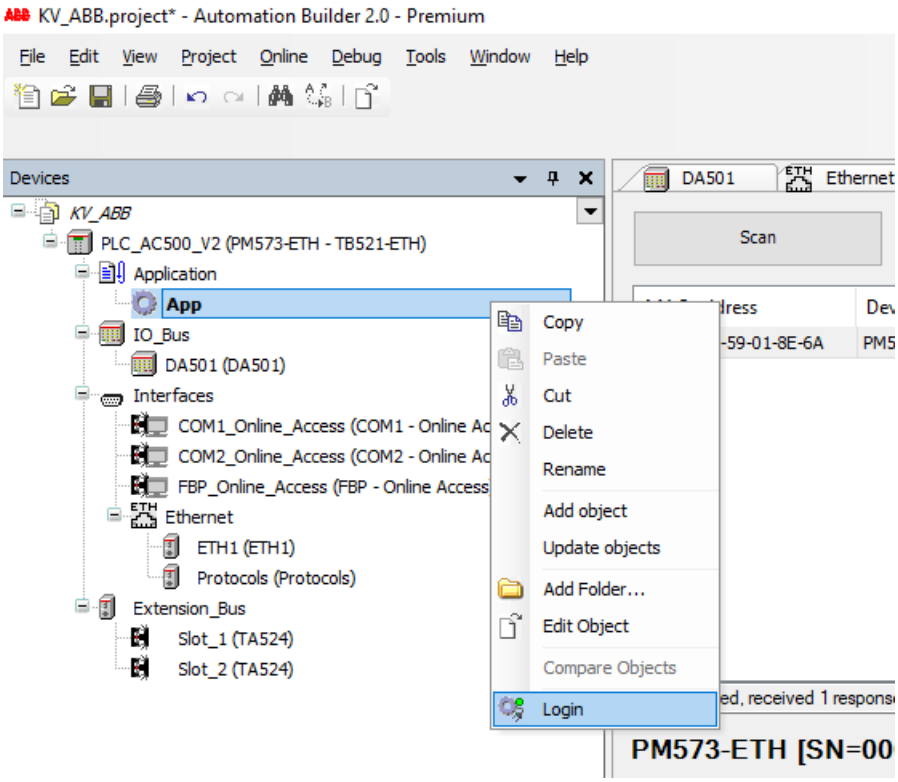

Figure 13: Login

If connection is successful App will be colored with a green.

Logout from the PLC.

After that step we can proceed to programming environment. In the Devices tree double click on Application--CoDeSys programming environment will run.

3 Project

The following objects are included in a project: POUs, Data types, Visualizations, Resources.

POU (Program Organization Unit). Functions, function blocks, and programs are POUs which can be supplemented by actions.

Each POU consists of a declaration part and a body. The body is written in one of the IEC programming languages which include IL, ST, SFC, FBD, LD or CFC. CoDeSys supports all IEC standard POUs.

In this course we are interested get to know Structured Text (ST). The Structured Text consists of a series of instructions which are determined as in high level languages.

Attention: Do not delete or rename the POU PLC_PRG (assuming you are not using a Task Configuration)! PLC_PRG is generally the main program in a single task program.

3.1 Monitoring

In Online mode, all displayable variables are read from the controller and displayed in real time. You will find this display in the declarations and program editor; you can also read out current values of variables in the watch and receipt manager and in a visualization. If variables from instances of function blocks are to be monitored, the corresponding instance must first be opened.

3.2 Simulation

During the simulation the created PLC program is not processed in the PLC, but rather in the computer on which CoDeSys is running. All online functions are available. That allows you to test the logical correctness of your program without PLC hardware.

NB! **As PLC does not have installed battery all the information downloaded to the controller will be lost when it is turned off.**

3 Binary variables

Value of the binary variable can be presented by: 1/0, ON/OFF, True/False.

Example 1 *Working with (physical) Global Variables*

In order to check what functions, blocks, variables and operations are available, you need to call Input Assistant}. In Program Window press **F2** button.

In Global Variables

- select any input address (S0-S7);
- select any digital output variable.

your_output:=your_input;

Download program to the controller

- in Online menu select Login or press <**ALT**>+<**F8**>;
- to Start program it should be in running mode: in Online menu select Run or **F5**.

If Program is not running (you cannot switch on Running mode), please reset error messages. Check the results using physical inputs of the PLC.

Example 2 *Combinatory logic*

Realize two logical functions $F = A \vee B\&C$ and $W = (A \vee B)\&C$ there logical operations are denoted as & -"AND" and $\vee$ - "OR".

Select any digital inputs (S0-S7) as A, B, C, and observable digital outputs: F, W.

Example 3 *Majority rule*

If you would like to comment your previous code, write it as (* commented code *).

Combine circuit of 3 inputs and one output: output is "ON" if at least two inputs are "ON".

Example 4 *Motor control*

START and STOP input signals turn on and off signals of the motor.

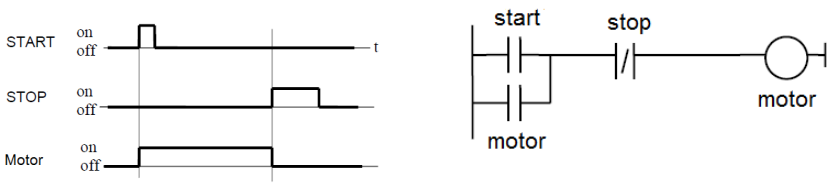

Figure 14: Single phase motor control

Select the controller inputs and outputs, tag them: START, STOP and Motor.

Write a program. Run program on PLC.