

APPLICATION NOTE

CONVERTING AN AC500 V2 PROJECT TO AN AC500 V3 PROJECT GUIDES, HINTS AND TIPS



Contents

1 Introduction					
	1.1	Scope of	the document	5	
	1.2	Compati	bility	6	
2	Main	changes f	rom AC500 V2 to AC500 V3	7	
-	2.1	-	e and system overview		
	L.1	2.1.1	Hardware and Operating System		
		2.1.2	Memory Sizes		
		2.1.3	Input- / output addresses		
		2.1.3	Memory allocation / byte order		
		2.1.4	Addressable variables		
		2.1.5	Modbus addresses / Byte order		
		2.1.0	Interfaces and slots		
	2.2		ion Builder V3 Editor		
	۲.۲	2.2.1	Configuration and programming in the same tool		
		2.2.1	Devices and POUs tree		
	2.3		nanagement and versioning		
	2.3 2.4	-	tion		
	2.4 2.5		S		
	2.5	2.5.1			
		2.5.1	Online diagnosis within Automation Builder		
	2.6		Diagnosis in IEC applications		
		-			
	2.7		nagement		
	2.8	2.8.1	al AC500 V3 capabilities		
			Object oriented programming (OOP)		
		2.8.2	Static code analysis (SCA)		
	20	2.8.3	Subversion control (SVN)		
	2.9		ntation of POUs and libraries		
		2.9.1	Documentation of the POUs inside the library manager		
		2.9.2	System technology about special libraries		
		2.9.3	Online Web Library's Documentation	.23	
3	Hardy	ware, prot	cocols and libraries availability	.25	
	3.1	Hardwar	e	25	
		3.1.1	CPU overview	25	
		3.1.2	CPUs detailed HW feature lists	26	
		3.1.3	Safety CPUs	34	
		3.1.4	AC31-Replacement	34	
		3.1.5	Hardware Interfaces	34	
		3.1.6	Fieldbus communication modules (CM)	34	
		3.1.7	Communication Interface modules	35	
		3.1.8	Other devices	35	
		3.1.9	Memory Card	36	
	3.2	Protocols	S	36	
		3.2.1	Serial protocols	36	
		3.2.2	Ethernet based protocols	.37	
		3.2.3	SMTP protocol	38	
	3.3	Libraries	·		
		3.3.1	AC500 System libraries	39	
		3.3.2	Communication libraries		
		3.3.3	Safety libraries		
		3.3.4	Application / Product libraries		
		3.3.5	CAA libraries		

		3.3.6	System libraries	43
		3.3.7	CODESYS libraries	
		3.3.8	Application libraries	44
		3.3.9	C-code libraries	44
4	Conv	version of	custom libraries	
•	4.1		e to V3.5	
	4.2		s in POUs, Libraries,	
	4.3	•	nd share the custom V3 library	
_	~		·	
5			the project	
	5.1	• •	adaptions in the Hardware configuration	
		5.1.1		
		5.1.2	Alarms, Traces & Recipes	
		5.1.3 5.1.4	Library Manager SNTP	
	5.2	·		
	5.2	5.2.1	adaptions in the IEC application	
		5.2.1 5.2.2	PLCopen conform function blocks Modbus addresses %M and %R	
		5.2.2 5.2.3		
	5.3		ACS/DCS Drives Library migration for 3 rd party devices	
	5.3	турісаі	migration for 3 party devices	
6	Rees	tablishin	g communication to field devices	59
	6.1	Modbus	s TCP	59
		6.1.1	AC500 V2	59
		6.1.2	AC500 V3	
		6.1.3	Summary of steps for migration from AC500 V2 to AC500 V3	
	6.2		s RTU	
		6.2.1	Modbus RTU HW configuration	63
		6.2.2	Modbus RTU IEC program adaption	65
	6.3		t	
		6.3.1	General	
		6.3.2	DriveManager	
		6.3.3	Bus cycle	
		6.3.4	Cyclic data exchange	
		6.3.5	Acyclic data exchange	
		6.3.6	Bus control	
		6.3.7	Diagnosis	
	6.4		S	
		6.4.1	General	
		6.4.2	DriveManager	
		6.4.3	Bus cycle	
		6.4.4	Cyclic data exchange	
		6.4.5	Acyclic data exchange	
		6.4.6	Bus control	
	a –	6.4.7	Diagnosis	
	6.5	CS31 Bu	is connections	76
7	Rees	tablishin	g communication to SCADA devices	78
	7.1		ح File	
		7.1.1	AC500 V2 PLC	
		7.1.2	AC500 V3 PLC	81
	7.2	CP600.		83
		7.2.1	Renaming tags e.g. CoDeSys V2 ETH to CODESYS V3 ETH	
			protocol	83

	7.2.2	Replacing tags e.g. ABB Modbus TCP V2 to ABB Modbus TCP	
		V3	
7.3	OPC DA		
	7.3.1	OPC DA for AC500 V2	90
	7.3.2	OPC DA for AC500 V3	94
	7.3.3	Update the project from AC500 V2 to AC500 V3	98
7.4	UDP		
7.5	MQTT & 3	JSON	

1 Introduction

1.1 Scope of the document

This document guides a user how to upgrade an existing AC500 V2 project to the AC500 V3 PLC generation. This document is split into the following chapters:

Main changes from AC500 V2 to AC500 V3

Overview of main changes between the AC500 V2 and AC500 V3 platforms. Most of these changes are concerning the hardware, e.g. different interfaces, different memory layout, different handling of addresses, etc.

• Hardware, protocols and libraries availability

The scope of communication modules, fieldbus protocols, libraries, etc. that are supported by AC500 V3 is constantly increasing. This chapter shall provide an overview on what is currently available, so it is easy to decide, if a certain application can be fully converted to AC500 V3.

In case something is missing in the AC500 V3 PLC, please get in touch with us, probably it is already on the roadmap.

• Conversion of custom libraries

Many AC500 V2 projects are using custom libraries, which could be either own developed libraries or libraries from other sources. The recommended first step before migrating the project is to either migrate the required AC500 V2 libraries (source code has to be available) or to check for AC500 V3 compatible versions of the libraries.

• Conversion of the project

Automated conversion of a project from AC500 V2 to AC500 V3 typically leads to some errors. In this chapter typical errors and recommended fixes are listed.

• Reestablishing communication to field devices

In most cases some dedicated manual adaptions have to be done for re-establishing communication to field devices. In this chapter typical errors and recommended adaptations to fix the errors are listed.

• Reestablishing communication to SCADA devices

In some cases dedicated manual adaptions have to be done for re-establishing communication to SCADA devices. In this chapter typical errors and recommended adaptations to fix the errors are listed.

When migrating from AC500 V2 to AC500 V3, security aspects should also be considered, e.g. by introduction of user management or by changing OPC DA communication to OPC UA communication.

1.2 Compatibility

The descriptions and screenshots in this document are based on an Automation Builder 2.5 installation.

In earlier versions some features described in this document are not available.

In newer versions the described steps might be partly different.

Automation Builder V2.6

Automation Builder versions can be installed side-by-side starting from version 2.6.0. The different versions are installed in separate installation folders.

For technical reasons there are some shared components across the different Automation Builder versions. Even despite the side-by-side installation of Automation Builder versions, these components will continue to be shared. Examples of shared components are AC500 V2 libraries, CODESYS V2 engineering and gateway, Panel Builder or Drive composer pro.

Limitations of Automation Builder 64bit:

- Library migration from AC500 V2 to AC500 V3 can only be triggered with an Automation Builder 32bit but not in the 64bit version
- Drive Manager for AC500 V2 is not available with Automation Builder 64bit

2 Main changes from AC500 V2 to AC500 V3

2.1 Hardware and system overview

This chapter contains a general overview of hardware capabilities of AC50 V3 compared to AC500 V2.

2.1.1 Hardware and Operating System

	AC500 V2	AC500 V3
Processor type	Power PC	TI ARM
Processor speed	Up to 400 MHz	Up to 1 GHz
Memory	Up to	Up to 160 MB
	• 4MB Code	(Code, data, %I area, %Q
	• 4MB Data	area, configuration, IEC driv- ers)
	• 32k %l	
	• 32k %Q	
	• 2MB Config	
M(odbus) memory	Up to 8 Segments with 64k each	128kB
Persistent memory	Up to 8 Segments %R with 64k each	128kB for persistent retain variables
Operating System	SMX	Linux
IO Bus	Up to 10 S500 devices	Up to 10 S500 devices
Ethernet	0-2 onboard Ethernet ports	1 or 2 onboard Ethernet ports

2.1.2 Memory Sizes

2.1.2.1 AC500 V2

	PM5x4	PM5x6	PM573	PM583	PM585	PM590	PM591	PM592	PM595
User Prog. Memory	128kB	512kB	512kB	1 MB	1 MB	2 MB	4 MB	4 MB	16 MB
User Data	10kB	64kB	224kB	736kB	1536kB	1536kB	4096kB	4096kB	16 MB
%M	2kB	64kB	128kB	128kB	512B	512kB	512kB	512kB	1024kB
VAR-Retain	1kB	1kB	32kB	32kB	512B	512kB	512kB	512kB	1024kB
%R	1kB	1kB	128kB	128kB	512B	512kB	512kB	512kB	1024kB
Total Variables	14kB	130kB	512kB	1024kB	3072kB	3072kB	5632kB	5632kB	19456kB
SRAM Disk	-	-	32 kB	64 kB	256 kB	256 kB	256 kB	256 kB	960 kB
Flash	1024kB	1512kB	3072kB	4096kB	12288kB	12288kB	16384kB	16384kB	40960kB
User RAM Disk	912kB	1424kB	1424kB	4096kB	4096kB	8 MB	8 MB	8 MB	32 MB
Flash Disk								4 GB	4 GB

2.1.2.2 AC500 V3

	PM5012	PM5032	PM5052	PM5072	PM5630	PM5650	PM5670	PM5675
User Prog. Memory	256 kB	512 kB	768 kB	1 MB	2 MB	8 MB	32 MB	32 MB
User Data								
Max. Size Download (Code, Data, Conf., WebVisu, Symbols)	1 MB	5 MB	7 MB	9 MB	9 MB	84 MB	176 MB	176 MB
%M	4 kB	16 kB	16 kB	64 kB	128 kB	128 kB	512 kB	512 kB
VAR-Retain	4 kB	16 kB	16 kB	36 kB	128 kB	128 kB	1024 kB	1024 kB
User Flash	30 MB	285 MB	643 MB	643 MB				
Total Flash	128 MB	512 MB	1024 MB	1024 MB				
Flash Disk								8 GB

2.1.3 Input- / output addresses

2.1.3.1 AC500 V2

IEC-Variable	Belegung
%IB0 - %IB999	I/O-Bus
%IB1000 - %IB1999	COM1 (CS31-Bus)
%IB2000 - %IB2999	COM2
%IB3000 - %IB3999	FBP-Interface
%IB4000 - %IB4095	Onboard-I/O (AC500-eCo)
%IB0.0 - %IB0.4095	Slot 0 (Interner CM module)
%IB1.0 - %IB1.4095	Slot 1 (Externer CM module1)
%IB2.0 - %IB2.4095	Slot 2 (Externer CM module2)
%IB3.0 - %IB3.4095	Slot 3 (Externer CM module 3)
%IB4.0 - %IB4.4095	Slot 4 (Externer CM module4)

All Inputs and Outputs are flat and Byte oriented

CM Modules: Inputs and Outputs with Slot Offset as Prefix

2.1.3.2 AC500 V3

One flat Address Range for

- IO-Bus
- Onboard CAN
- COM1
- CM Module

Will be filled without Gaps

2.1.4 Memory allocation / byte order

AC500 V2	AC500 V3
Big Endian	Little Endian
(Motorola byte order)	(Intel byte order)

2.1.4.1 AC500 V2

ADR	adr	adr+1	adr+2	adr+3	
BOOL	%IX0.x	%IX1.x	%IX2.x	%IX3.x	
BYTE	%IB0	%IB1	%IB2	%IB3	
WORD	%	W0	%	W1	
DWORD		%	DO		
%IX0.0	:=TRUE				
%IB0	:=16#01	:=1			
%IW0	:=16#0100	:=25	:=256 (Bit 8)		
%ID0	:=16#0100000	00 :=16	5777216		
%IX3.0	:=TRUE				
%IB3 :=16#01		:=1			
%IW1	:=16#0001	:=1	(Bit 0)		
%ID0	:=16#000000	01 :=1			

2.1.4.2 AC500 V3

ADR	adr	adr+1	adr+2	adr+3	
BOOL	%IX3.x	%IX2.x	%IX1.x	%IX0.x	
BYTE	%IB3	%IB2	%IB1	%IB0	
WORD	%I	W1	%	W0	
DWORD		%	DO		
%IX0.0	:=TRUE				
%IB0	:=16#01	:=	=1		
%IW0	:=16#0100	:=	=1 (Bit 0)		
%ID0	:=16#00000	.=	=1 (Bit 0)		
%IX3.0	:=TRUE				
%IB3	:=16#01	:=	=1		
%IW1	:=16#0001	:=	:=1 (Bit 8)		
%ID0	:=16#010000	:=: 000	= 16777216		

2.1.5 Addressable variables

2.1.5.1 AC500 V2

• %M- and %R- Area (4k - 512k depending on CPU type)

• Max 8 Segments with 64kB

Linie	IEC - Variable
0	%MB0.0 %MB0.65535 %RB0.0 %RB0.65535
1	%MB1.0 %MB1.65535 %RB0.0 %RB0.65535
6	%MB6.0 %MB6.65535 %RB0.0 %RB0.65535
7	%MB7.0 %MB7.65535 %RB0.0 %RB0.65535

• Bits are Byte oriented

ADR	adr	adr+1	adr+2	adr+3
BOOL	%MX0.0. 0 7	%MX0.1. 0 7	%MX0.2. 0 7	%MX0.3. 0 7
ВҮТЕ	%MB0.0	%MB0.1	%MB0.2	%MB0.3
WORD	%MW0.0 %MW0.			W0.1
DWORD	%MD0.0			

2.1.5.2 AC500 V3

- %M, one common Area (4-512 kB see chapter 2.1.2.2)
 - o %MB0...%MB131071
- Bits are Byte oriented
- Persistent Option: %M without init (pragma: {no init}) to be set at each variable
- No %R Area, only VAR GLOBAL PERSISTENT RETAIN

Details for converting Memory ranges from V2 to V3 can be found in chapter 5.2.2.

2.1.6 Modbus addresses / Byte order

Due to the internal different memory layout, also the Modbus addressing has changed from AC500 V2 to AC500 V3. There are no more different segments, but only one flat memory area.

2.1.6.1 AC500 V2

Detailed information about AC500 V2 Modbus addresses is available from the online help:

PLC Automation with V2 CPUs \rightarrow PLC integration (hardware) \rightarrow System technology for AC500 V2 products \rightarrow System technology of CPU and overall system \rightarrow Communication with Modbus RTU \rightarrow Modbus addresses for AC500 CPUs \rightarrow Modbus address table

Address assignment (bit accesses)

The address assignment for bit accesses is done according to the following table:

Modbus address		Byte	Bit (byte-oriented)	Word	Double word	
HEX	DEC	BYTE	BOOL	WORD	DWORD	
Line 0						
0000	0	%MB0.0	%MX0.0.0	%MW0.0	%MD0.0	
0001	1		%MX0.0.1			
0002			%MX0.0.2			
0003	3		%MX0.0.3			
0004	4		%MX0.0.4			
0005	5		%MX0.0.5			
0006	6		%MX0.0.6			
0007	7		%MX0.0.7			
0008	8	%MB0.1	%MX0.1.0			
0000	0		0/14/0 4 4			
001E	30		%MX0.3.6			
001F	31		%MX0.3.7			
0020	32	%MB0.4	%MX0.4.0	%MW0.2	%MD0.1	
0021	33		%MX0.4.1			
0022	34		%MX0.4.2			
OFFF	4095	%MB0.511	%MX0.511.7	%MW0.255	%MD0.127	
1000	4096	%MB0.512	%MX0.512.0	%MW0.256	%MD0.128	
7FFF	32767	%MB0.4095	%MX0.4095.7	%MW0.2047	%MD0.1023	
8000	32768	%MB0.4096	%MX0.4096.0	%MW0.2048	%MD0.1024	
FFFF	65535	%MB0.8191	%MX0.8191.7	%MW0.4095	%MD0.2047	

Calculation of the bit variable from the hexadecimal address:

Formula:						
	Bit variable (BOOL) := %MX0.BYTE.BIT					
where:	DEC	Decimal address				
	BYTE	DEC / 8				
	BIT	DEC mod 8	(Modulo division)			

2.1.6.2 AC500 V3

Detailed information about AC500 V3 Modbus addresses is available from the online help:

PLC Automation with V3 CPUs \rightarrow PLC integration (hardware) \rightarrow System technology for AC500 V3 products \rightarrow System technology of CPU and overall system \rightarrow Communication with Modbus RTU \rightarrow Modbus addresses for AC500 V3 processor modules PM56xx \rightarrow Modbus address table

Mo	odbus address	Byte Bit (byte-oriented)		Word	Double word	
HEX	DEC	BYTE	BOOL	WORD	DWORD	
0000	0	%MB0	%MX0.0 %MX0.7	%MW0	%MD0	
		%MB1	%MX1.0 %MX1.7			
001	1	%MB2	%MX2.0 %MX2.7	%MW1		
		%MB3	%MX3.0 %MX3.7			
0002	2	%MB4	%MX4.0 %MX4.7	%MW2	%MD1	
		%MB5	%MX5.0 %MX5.7			
0003	3	%MB6	%MX6.0 %MX6.7	%MW3		
		%MB7	%MX7.0 %MX7.7			
FE	65534	%MB131068	%MX131068.0 %MX131068.7	%MW65534	%MD32767	
		%MB131069	%MX131069.0 %MX131069.7			
FF	65535	%MB131070	%MX131070.0 %MX131070.7	%MW65535		
		%MB131071	%MX131071.0 %MX131071.7			

CONVERTING AN AC500 V2 PROJECT TO AN AC500 V3 PROJECT

able 623: Address assignm		Protection	Rit (hute exists 1)	Mard	Dauble we-t
	Modbus address	Byte BYTE	Bit (byte-oriented) BOOL	Word WORD	Double word DWORD
HEX	DEC				
0000	0	%MB0	%MX0.0	%MVV0	%MD0
0001	1		%MX0.1		
0002	2		%MX0.2		
0003	3		%MX0.3		
1004	4		%MX0.4		
005	5		%MX0.5		
006	6		%MX0.6		
007	7		%MX0.7		
800	8	%MB1	%MX1.0		
1009	9		%MX1.1		
00A	10		%MX1.2		
100B	11		%MX1.3		
00C	12		%MX1.4		
100D	13		%MX1.5		
00E	14		%MX1.6		
100F	15		%MX1.7		
010	16	%MB2	%MX2.0	%MW1	
1011	17		%MX2.1		
012	18		%MX2.2		
0020	32	%MB4	%MX4.0	%MW2	%MD1
0021	33		%MX4.1		
0022	34		%MX4.2		
0FFF	4095	%MB511	%MX511.7	%MW255	%MD127
1000	4096	%MB512	%MX512.0	%MW256	%MD128
7FFF	32767	%MB4095	%MX4095.7	%MW2047	%MD1023
8000	32768	%MB4096	%MX4096.0	%MW2048	%MD1024
FFFF	65535	%MB8191	%MX8191.7	%MW4095	%MD2047

Calculation of the bit variable from the hexadecimal address:

Formula:					
	Bit variable (BOOL) := %MXBYTE.BIT				
where:	DEC	Decimal address			
	BYTE	DEC / 8			
	BIT	DEC mod 8	(Modulo division)		

2.1.7 Interfaces and slots

AC500 V2	AC500 V3			
0-2 Onboard Ethernet	1-2 Onboard Ethernet			
2 COM Ports (incl. CS31-Bus Master at COM1)	1 COM Port			
	1 CAN Onboard Port			
1 FBP Port				
Memory card slot	(Micro) memory card slot			
Battery Slot	Battery Slot			
Up to 4 external communication modules	Up to 6 external communication modules			

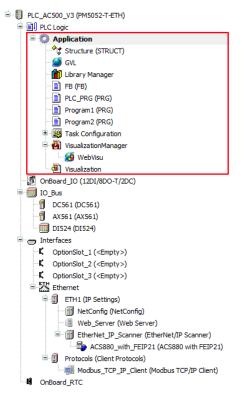
Device name will be handled by Automation Builder during inserting the communication module into the tree

2.2 Automation Builder V3 Editor

2.2.1 Configuration and programming in the same tool

AC500 V2	AC500 V3		
Configuration in Automation Builder	Configuration in Automation Builder		
Programming in CODESYS	Programming in Automation Builder		

In contrast to AC500 V2, where only the configuration is done in Automation Builder and the programming is done in CODESYS, the AC500 V3 PLCs are configured and programmed in Automation Builder.



2.2.2 Devices and POUs tree

In Automation Builder there is a device tree and a POU tree. In AC500 V2 only the device tree is used.



In the device tree several AC500 V2, AC500 V3 or other devices like Panels, Drive composer pro, ... can be added. All devices have an own application containing variables, structures, programs, ...

In the POU tree there can be also global variables, function blocks, structures, visualizations, libraries, ... These POU objects can be used by all AC500 V3 PLCs in the device tree.

2.3 Library management and versioning

AC500 V2	AC500 V3
Only one library version for all Automation Builder Versions	Different library versions depending on Au- tomation Builder Version. Multiple library versions within one Profile possible
Libraries stored in folders	Libraries installed to library repository
Library manager in CODESYS 2.3 Project	Library Manager in Automation Builder de- vice tree and optional in the POU tree

In AC500 V3 the library manager is in the device tree below the application and if needed inside the POU tree. For more details between the device tree and POU tree see chapter 2.8.

Necessary libraries required for the hardware functionality are automatically added to the library manager depending on the used hardware configuration in the device tree.

Additional libraries can be added from the *Library Manager* object by clicking on *Add Library* (1).

The libraries can be shown in a tree view or as list (2)

The System libraries are by default hidden. To make them visible click Display advanced libraries in the upper right corner. (3)

/	👔 Library Manager 🗙	
E	Add Library 1 elete Library Properties 🗃 Details 🗐 Place	ceholders 🛛 🎁 Library Repository 🕕 Icon Le
A	dd Library	2 3
	String for a fulltext search	
	Library	Company
	ABB - AC500 Application Definition Definition	
	Use Cases	

After selecting the library from the list, it can be added by clicking *OK*.

In contrast to V2 a library repository is handling all libraries and versions. To add an external library to a project it first needs to be installed in the library repository.

In AC500 V3 a versioning inside the library repository is possible. When adding a library to a project, the placeholder (e.g. AC500_Ethernet) is added. The device resolves this placeholder via the library repository resolves with the matching version of this library (e.g. Ethernet 1.3.0.7). It is also possible to change the used version by clicking Placeholders and double clicking the library which shall be changed. If multiple versions of this library are installed to the repository the version can be changed from the placeholder here.

SysTime	SysTime, 3.5.17.0 (System)
SysTimeCore SysTimeRtc VisuDialogs	Other versions of SysTime 3.5.9.0 3.5.5.0
<	3.2.0.0 Other library

It is not recommended to use older versions of a library, if multiple versions are installed.

2.4 Visualization

The AC500 V3 visualization provides more options in terms of configuration and widgets than AC500 V2. Main difference of the web visualization is that the generated visualization is plain HTML 5 and does not require Java Script. Detailed descriptions of the V3 features are available in the online help and the application example: AC500 V3 Web Visualization – Demonstration Example

2.5 Diagnosis

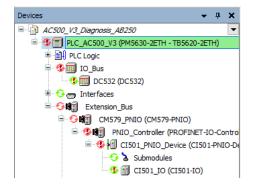
Diagnosis for commissioning or maintenance is provided either by Automation Builder or IEC code. While AC500 V2 only shows generic diagnosis information, AC500 V3 is able to receive device-, module- or channel specific diagnosis.

2.5.1 Online diagnosis within Automation Builder

Automation Builder diagnosis for AC500 V2 is only available as generic CPU diagnosis at the PLC node.

evices 👻 🖣	×	PLC_AC500_V2 X						
CPU ActiveErrors AppExample AB251	•	CPUDiagnostics		Read Errors			Acknowledge	
b) Application_1 App [run]		Statistics						
IO_Bus		Version information	Index	State	Ack.	Class	Description	Online text
DC532 (DC532)			●! 0	Active	No	E4	Battery status changed	E4:
- III DI524 (DI524)		PLC Browser	• 1	Active	No	E4	Battery is missing or empty	E4:
DO562 (DO562)			•! 2	Active	No	E4	SRAM disk has been formatted	E4:
- 🗐 AI531 (AI531)		PM583-ETH Parameters	•! 3	Inactive	No	E4		E4:
AX521 (AX521)			•! 4	Active	No	E3	Invalid configuration at Component/Device	E3: I/O-Bus, Mod.
D0561 (D0561)		PM583-ETH Hardware	●! 5	Active	No	E3	Invalid configuration at Component/Device	E3: I/O-Bus, Mod.
DA501 (DA501)			•! 6	Active	No	E3	Invalid configuration at Component/Device	E3: I/O-Bus, Mod
		I/O mapping list	e! 7	Active	No	E3	Invalid configuration at Component/Device	E3: I/O-Bus, Mod.
🖬 💭 COM1_Online_Access (COM1 - Online Access)		Information	•! 8	Active	No	E3	Invalid configuration at Component/Device	E3: I/O-Bus, Mod.
🛱 💭 COM2_Online_Access (COM2 - Online Access)		Information	®! 9	Active	No	E3	Invalid configuration at Component/Device	E3: I/O-Bus, Mod.
FBP_Online_Access (FBP - Online Access)			10	Active	No	E3	Invalid configuration	E3: I/O-Bus
🖻 🚟 Ethernet			• 11	Inactive	No	E4	Timeout while updating the I/O data at program start	E4: I/O-Bus
ETH1 (ETH1) Protocols (Protocols)								

Compared to AC500 V2, the online view of Automation Builder for AC500 V3 PLCs provides first indication of faults and errors at the device tree in form of symbols in front of the faulty device and their parent objects.



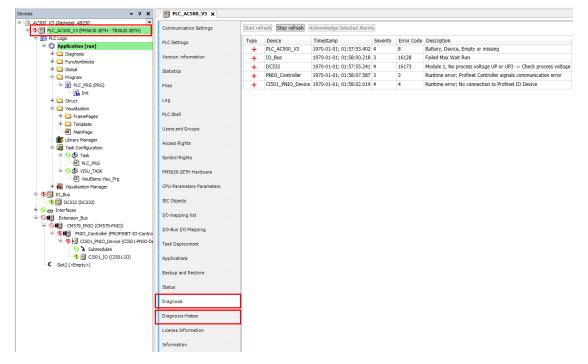
Note: Read more about *Project tree in online mode* for AC500 V3 in the Automation Builder online Help. For this, follow the content path:

PLC Automation with V3 CPUs > Diagnosis and debugging for AC500 V3 products > Online diagnosis in Automation Builder > Project tree in online mode

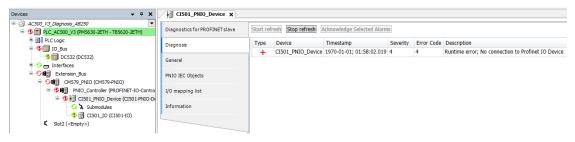
For AC500 V3 not only generic diagnosis, but also device and feature specific diagnosis is available inside Automation Builder during online mode.

The PLC provides a general overview of all diagnosis information inside the system. The tab "Diagnosis" shows only actual pending errors and faults and therefore has only incoming diagnosis data.

To see the history of incoming and outgoing diagnosis data, AC500 V3 has also a tab "Diagnosis History".



Next to the overview of the PLC node, the diagnosis is also available at their specific device node.





(hi)

Note: Read more about *Device diagnosis* and *Diagnosis history* for AC500 V3 in the Automation Builder online Help. For this, follow the content path:

PLC Automation with V3 CPUs > Diagnosis and debugging for AC500 V3 products > The diagnosis system > Diagnosis in Automation Builder

2.5.2 Diagnosis in IEC applications

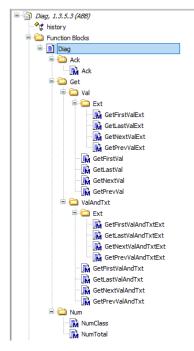
AC500 V2	AC500 V3
Generic diagnosis error codes	Device specific diagnosis error codes
-	Diagnosis available as text
-	Diagnosis history available for IEC applica- tions
-	Editable text lists available to modify reada- ble text
-	Multi language system available

To receive diagnosis data in IEC applications, a library is required for AC500 V2 as well as for AC500 V3:

- AC500 V2: Diag_AC500_V20
- AC500 V3: AC500_Diag

AC500_DiagHistory (not available for V2)

For V3 platform, the library uses methods to read out diagnosis data. The diagnosis of the complete system can be read with the function block *Diag* inside the library.

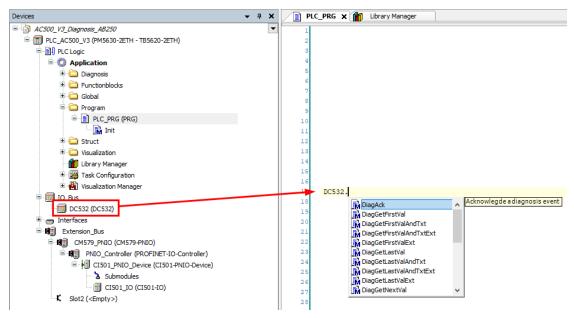


In order to receive diagnosis for specific devices, the checkbox *Enable diagnosis for devices* must be activated at the PLC Settings tab of the PLC node.

Devices 👻 🕂 🗙	PLC_AC500_V3 X		
	Communication Settings	Application for I/O handling Application V	
■ 留川 PLC Logic ■ ③ Application	PLC Settings	-PLC Settings	
8년 🏠 Diagnosis 8년 🚔 Functionblocks	Version information	Behavior for outputs in stop Set all outputs to default \checkmark	
🖶 🧰 Global 🕸 💼 Program	Statistics	Always update variables Disabled (update only if used in a task) V	
🗟 Struct 🗷 🧰 🧰 Visualization	Files	Bus cycle Options Bus cycle task (unspecified) 	
🖬 Library Manager 🖻 🎉 Task Configuration	Log	Additional Settings	
🗷 🍓 Visualization Manager	PLC Shell	Generate force variables for IO mapping 🗹 Enable diagnosis for devices	
□ ■ 10_Bus □ = ■ DC532 (DC532)	Users and Groups	Show I/O warnings as errors	

If this is done device specific diagnosis can be read, by accessing the methods for diagnosable device object inside the device tree.

For this, the object name of the device tree must be used as shown in the following picture.



Note: Read more about *Diagnosis in IEC applications* for AC500 V3 in the Automation Builder online Help. For this, follow the content path:

PLC Automation with V3 CPUs > Diagnosis and debugging for AC500 V3 products > The diagnosis system > Diagnosis in IEC application

hi,

Note: An application example on how to use AC500 V3 diagnosis inside IEC applications can be found here.

An application example for the diagnosis history can be found here.

2.6 Security

In general, please have a look at our whitepaper. This document provides a good overview about the topic Hardening and Defense in Depth strategy.

It is strongly recommended that no devices should be connected to the internet without additional security measures like encrypted communication or use of VPN. An application Note: Secure remote access via secomea gateway will explain one possibility.

2.7 User Management



Note: Since the user managements in AC500 V2 and AC500 V3 are completely different. The user management cannot be upgraded and must be re-done in AC500 V3.

Automation Builder provides a feature to protect the project file with a password or certificate. Please have a look at the application note: AC500 USER MANAGEMENT WITH V3. This feature is available for AC500 V2 and AC500 V3 project files.

In addition, the AC500 V3 provides two further user managements for:

- PLC
- Visualization

Further details can be found in the application note above or in the Automation Builder online help: PLC Automation with V3 CPUs > Programming with CODESYS > CODESYS Development System > Downloading an Application to the PLC > Handling of Device User Management

AC500 V2 PLC's user management can be accessed inside CODESYS under: Project \rightarrow "User groups and passwords...". Here different passwords for different user groups (Level 0 ... Level 7) can be set.

Level 0 is admin level.

Password for user group		×
User Group:	Password:	ОК
Levero	Confirm Password:	Cancel

Once the passwords are set, you can apply this user rights to different objects. For example, the PLC_PRG. Right click on PLC_PRG \rightarrow Object Properties

💊 CoDeSys - Application.AC500PRO									
File Edit Project Insert Extras Online Window	Help								
▋▋▋▋▓▓▋▙▌▙▓▓▌▓▐▙									
a POUs	1							7	×
PLC_PRG (PRG)	Properties							f	~
	Access rights								
	User Groups	0	1	2	3	4	5	6	7
	No Access	С	С	С	С	С	С	С	0
	Read Access	$^{\circ}$	С	С	С	С	$^{\circ}$	С	0
	Full Access	•	•	•	ſ	ſ	ſ	•	•
	C Apply to all								
							ОК		Cancel

The following screen will be shown. Here you can set access to different permissions for the user groups.

2.8 Additional AC500 V3 capabilities

When migrating a project from AV500 V2 to AC500 V3, it should be considered to combine that migration activity with a refactoring of the application to reduce potential technical debt.

Besides that the following AC500 V3 capabilities are available and should be considered to implement.

2.8.1 Object oriented programming (OOP)

Information is available in the online help. See: PLC Automation with V3 CPUs > Programming with CODESYS > CODESYS Development System > Programming of Applications > Object-Oriented Programming

Or in the Application Example AC500 V3 Object Oriented Programming - Working with Interfaces, Methods and Properties or in the Application Note AC500 V3 OOP Keywords

2.8.2 Static code analysis (SCA)

Information is available in the online help. See: PLC Automation with V3 CPUs > Engineering interfaces and tools > CODESYS Static Analysis

Or in the Application Example AC500 V3 Pro_Static_Analysis - Usage and Benefits For Code Optimization

2.8.3 Subversion control (SVN)

Information is available in the online help. See: PLC Automation with V3 CPUs > Engineering interfaces and tools > Professional Version Control

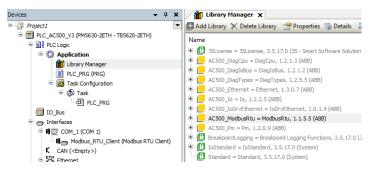
2.9 Documentation of POUs and libraries

2.9.1 Documentation of the POUs inside the library manager

The documentation of the functions, function blocks, structs etc (POU) for AC500 V3 are all included in the POUs and can be found directly in the library manager by opening the POU.

These informations are directly extracted from the POU source code, which ensures that they are always up to date.

Library Manger > dedicated library >



click on the POU > tabs on the right

2.9.1.1 Inputs/Outputs tab

All inputs and outputs are described with their type, inherited from, address, initial value and comment.

The inherited inputs and outputs are listed first. That means the order is not according to inputs than outputs.

AC500_ModbusRtu = ModbusRtu,	, 1.1.5.5 (ABB)		AC5	00_Modbu	sRtu	1.1.5.5	0
	AC5	00 Pm		1.2.6.9	0		
🖃 🛅 ModbusRtu, 1.1.5.5 (ABB) 💌	- Inputs/Outputs 🚮 Grap	ohical 🕐 Documenta	tion				
+ istory	FUNCTION_BLOCK ModR	UNCTION_BLOCK ModRtuToken EXTENDS AbbLConCA					
🖻 🚞 Function Blocks	Name	Туре	Inherited from	Addr	Initial	Comment	
ModRtuMast	🍫 Enable	BOOL	AC500 StateMachi		FALSE	A rising edge ((Enable = TRUE) starts the operation, the (
ModRtuRead	Susy	BOOL	AC500 StateMachi		FALSE	Operation is ru	unning (while output Error is FALSE)
ModRtuReadWrite23	🐶 Error	BOOL	AC500 StateMachi		FALSE	Operation is st	topped with error (while output Busy is FAL
ModRtuToken	牧 Com	BYTE			1	2: Modbus con	nmunication port - e.g: 1 = COM1 - change
ModRtuWrite	牧 Serv	BYTE			1	3: Modbus RTU	U server address - if changed while a modb
🖲 🧰 Types	牧 TimeOut	WORD			1000	4: Timeout [ms	s] for ModRtuMast function block - TimeOu
Visualization	ReconnectPause	WORD			0	5: Pause in se	conds before next retry to connect after a
GlobalTextList	🧐 LineToken	ModRtuTokenType				6: Reference v	variable to connect to other Modbus RTU f
	🦘 ServData	ModRtuGenDevDataT	2			7: Modbus RTU	U server reference variable to connect to a
	SerrorID	ERROR ID				4: Error codes	
	ModMastErrorAct	BOOL			FALSE	5: Active error	r in ModMast. Operation is running with err

In case of longer comment a tooltip will show the whole comment.

Operation is stopped with error (while output busy is FALSE). This output is TRUE for a least one cycle or unuit choice is set to FALSE. The output choice gives more		
2: Modbus communication port - e.g: 1 = COM1 - changes are valid only after rising edge of Enable input (FALSE -> TRUE). Valid values are 1 to 3 depending on PLC		
3: Modbule TU server address - if changed while a modbus job is running, this job will be finished with previous Serv address; 0 = Broadcast. valid range is 0247.		
4: Timeou 2: Modbus communication port - e.g: 1 = COM1 - changes are valid only after rising edge of Enable input (FALSE -> TRUE). Valid values are 1 to 3 deper	ding o	n PLC type and configuration.
5: Pause in seconds before next retry to connect after a timeout was detected. Timeout is detected with ModMastErrorID = 16#120		

2.9.1.2 Graphical tab

A graphical view of the POU shows all inputs, in_outputs and outputs including their type and default initial value.

- Inputs/Outputs 🚺 Graphical ?	Documentation
	dRtuToken
[Enable BOOL := FALSE] [Com BYTE := 1]	BOOL Busy BOOL Error
[Serv BYTE := 1] [TimeOut WORD := 1000] 	<i>ERROR_ID</i> ErrorID <i>BOOL</i> ModMastErrorAct <i>ERROR ID</i> ModMastErrorIDLast
LineToken ModRtuTokenType ServData ModRtuGenDevDataType	BOOL WarnAct WARNING ID WarnIDLast
Servoata mountusendevoatarype	BOOL JobBusy
	BOOL Online

2.9.1.3 Documentation tab

In the documentation tab all comment of the declaration part of the POU is shwon.

Depending on the POU, this can include a longer detailed description of how to use the POU, followed by the description of the inputs, in_outputs and outputs.

					0
dRtul	Token (FB)				E
CTION_I	BLOCK ModRtuToker	EXTENDS AbbLConC	A		
municati	on to generic Modbus	server devices via Mo	dbus RTU using Li	ineToken variable.	
ny field d	levice which supports		ithin it such as PL	generic Modbus RTU server device. A generic Modbus RTU server d C, HMI or ABB ACS/DCS drive etc. It must be used together with the change Modbus data.	
/DCS dri onnected	ve) function block mu I to the same LineTok	st be used. All these M	odRtuToken and/o RtuTokenType at	am an own instance of ModRtuToken or DrvModbusRtu (for connectic r DrvModbusRtu (for connection to an ABB ACS/DCS drive) function their IN_OUT LineToken. Via this LineToken variable the serial access to PI C task.	blocks must
tion block	k. The Modbus job wil		ModRtuToken func	s jobs are transferred via IN_OUT ServData variable to the ModRtuTe tion block of the server has the token, and the read/write function block task.	
onnectPa		steady delay for a disc		ut this server will not be reconnected for the time specified at the inpu an be avoided.	t
onnectPa or FRR C	OM DIFFFRENT I II	steady delay for a disc			ıt
onnectPa or FRR C	OM DIFFERENT I II	steady delay for a disc			
onnectPa or FRR C	OM DIFFFRENT I II	steady delay for a disc		an be avoided.	.t
onnectPa r FRR C	use in seconds. So a OM DIFFFRENT I II uts Graphical	steady delay for a disc NF Documentation	connected server c	an be avoided. details about the error. 2: Modbus communication port - e.g: 1 = COM1 - changes are valid only after rising edge of Enable input (FALSE -> TRUE). Valid values are 1 to 3 depending on PLC type and	
onnectPa	use in seconds. So a OM DIFFFRENT I II uts Graphical	Documentation	1	details about the error. 2: Modbus communication port - e.g: 1 = COM1 - changes are valid only after rising edge of Enable input (FALSE -> TRUE). Valid values are 1 to 3 depending on PLC type and configuration. 3: Modbus RTU server address - if changed while a modbus job is running, this job will be finished with previous Serv	

				details about the error.	G
	Com	BYTE	1	 Modbus communication port - e.g: 1 = COM1 - changes are valid only after rising edge of Enable input (FALSE -> TRUE). Valid values are 1 to 3 depending on PLC type and configuration. 	E
Input	Serv	ВУТЕ	1	 Modbus RTU server address - if changed while a modbus job is running, this job will be finished with previous Serv address; 0 = Broadcast. valid range is 0247. 	
	TimeOut	WORD	1000	4: Timeout [ms] for ModRtuMast function block - TimeOut value should be at least 50ms.	
	ReconnectPause	WORD	0	5: Pause in seconds before next retry to connect after a timeout was detected. Timeout is detected with ModMastErrorID = 16#120	
Inout	LineToken	ModRtuTokenType		6: Reference variable to connect to other Modbus RTU function blocks ModRtuRead, ModRtuWrite and ModRtuReadWrite23.	
	ServData	ModRtuGenDevDataType		7: Modbus RTU server reference variable to connect to all function blocks of this server device.	
	ErrorID	ERROR ID	-	4: Error codes	

System technology about special libraries 2.9.2

For some libraries special system technology docmentation can be found in the Automation Builder help. See: PLC Automation with V3 CPUs > Libraries and solutions

2 Help	
🔇 Back 💿 🛃 🛁 🛛 🌏 Contents 🔥 Index	🔍 Search @
Contents	4 X Libraries and solutions
PLC Automation PLC Automation portfolio How to in this document PLC Automation with V2 CPUs	PLC Automation with V3 CPUs > Libraries and solutions
UPLC Automation with V3 CPUs Cetting started Portunation Builder installation man Portunation Builder installation man Programming with CODESYS C() Libraries and solutions	Libraries and solutions
	 Information on libraries Reference to CODESYS (V3) Library Manager functionality ACS/DCS drives libraries BACnet-BC CAA library quidelines

2.9.3 **Online Web Library's Documentation**

In this Web online documentation all POUs are documented based on an export from the information in the library manager of Automation Builder (see 2.9.1).

On the left side a tree view can be opened and a search option is provided.

e to Library's documentation!
ne to Library's documentation!
500
ses
- Application
ser
IS

It's necessary to have the Automation Builder installed and open to use it.

Access to this Library's documentation can be found from

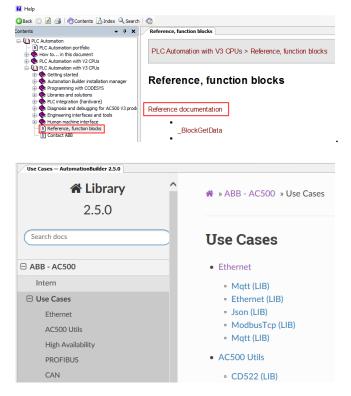
- Automation Builder > Help > Library web help

<u>H</u> el	р	
۲	<u>C</u> ontents	Ctrl+Shift+F1
?	Index	Ctrl+Shift+F2
	Search	
1	ABB Drives homepage	
1	ABB PLC homepage	
1	ABB <u>R</u> obotics homepage	
۸	Library web help	
		M.

_

From the Automation Builder Help via PLC Automation with V3 CPUs > Reference,

function blocks, then click on "Reference documentation"



3 Hardware, protocols and libraries availability

This chapter contains the information which hardware, protocols and libraries from AC500 V2 are ready to be used with AC500 V3 and which are not yet there available.



This chapter gives an overview but will not be updated regularly. As we are working with high priority to fill all gaps. This list might be outdated, and features might be available already, even if marked as not in here.

3.1 Hardware

3.1.1 CPU overview

The following tables give a rough overview of possible replacements, but the detailed HW feature list needs to be checked to find out, if all needed HW features are available.

AC500 CPUs

CPU type V2	CPU type V3	Remark
PM572, PM582	РМ5630-2ЕТН / РМ5650- 2ЕТН	Check features see 3.1.2.1
РМ573-ЕТН, РМ583-ЕТН	РМ5630-2ЕТН / РМ5650- 2ЕТН / РМ5670-2ЕТН	Check features see 3.1.2.1
PM585-ETH	РМ5630-2ЕТН / РМ5650- 2ЕТН / РМ5670-2ЕТН	Check features see 3.1.2.1
РМ590-ЕТН, РМ591-ЕТН	РМ5650-2ЕТН / РМ5670- 2ЕТН	Check features see 3.1.2.1
PM592-ETH	РМ5670-2ЕТН / РМ5675- 2ЕТН	Check features see 3.1.2.1
PM592-ETH CMS	Not yet available	Check features see 3.1.2.1
PM595-4ETH	РМ5670-2ЕТН / РМ5675- 2ЕТН	Check features see 3.1.2.1

AC500-eCo CPUs

CPU type V2	CPU type V3	Remark
PM554 / PM564	PM5012/ PM5032	Check features in 3.1.2.2
PM554-ETH / PM564-ETH	PM5032/ PM5052	Check features in 3.1.2.2
PM556-ETH / PM566-ETH	PM5052/ PM5072	Check features in 3.1.2.2



3.1.2 CPUs detailed HW feature lists

This chapter contains high level overview of the availability of AC500 V2 features on potential AC500 V3 substitute PLCs. A green background color means that the feature is available, wither with same or extended scope. A yellow background means that there might be issues in migrating the PLC application. There-fore, it is highly recommended to clarify the options with your ABB sales representative.

3.1.2.1 Overview feature list of AC500 V2 with potential AC500 V3 substitute PLCs

12	A B C	DE	F	G	Н	М	Ν
	1 Type		PM572	PM582(-XC)		PM5630-2ETH(-XC)	PM5650-2ETH(-XC)
	2						
	3 Supply voltage	Better					
	4 24 V DC	Slighty better	Yes	Yes		Yes	Yes
	5 100-240 V AC	Identical / Similar	No	No		No	No
	6 Type of processor / Processor clock frequency	Lower feature	Freescale ARM Processor 32-bit / 50 MHz	Freescale ARM Processor 32-bit /84 MHz		TI ARM Cortex-A9 32-bit-RISC / 300 MHz	TI ARM Cortex-A9 32-bit-RISC / 600 MHz
	7 Total RAM memory / Total Flash memory	Not available	32 MB / 16 MB	32 MB / 16 MB		128 MB / 128 MB	256 MB / 512 MB
	8 Total user program memory (3)		256 kB	928 kB		8 MB	80 MB
	20 Real-time clock		Yes	Yes		Yes	Yes
	22 Plug-in memory card		Yes	Yes		Yes	Yes
+	24 Program execution						
	35 Onboard digital inputs		No	No		No	No
	42 Onboard digital outputs		No	No		No	No
	56 Onboard digital input /output, configurable channels		No	No		No	No
	68 Onboard analog outputs		No	No		No	No
	72 Onboard analog inputs		No	No		-	-
	76 Max. number of centralized inputs/outputs						
	85 Max. number of decentralized inputs/outputs		depends on the used standard fieldbus	depends on the used standard fieldbus		depends on the used standard fieldbus	depends on the used standard fieldbus
+	88 Option board slots for extension		No	No		No	No
	99 Internal interfaces						
	00 COM1		Yes	Yes		Yes	Yes
+ 1	10 COM2		Yes	Yes		No	No
+ 1	19 FieldbusPlug		Yes, but FieldBusPlug is out phased	Yes, but FieldBusPlug is out phased		No	No
+ 1 + 1 + 1	26 CAN Interface		No	No		Yes	Yes
+ 1	36 Ethernet		-	-		Yes	Yes
+ 1	85 Terminal base for connection / CMS		Yes / No	Yes / No		Yes / No	Yes / No
	95 Supported communication with Communication Module						
	96 Serial communication based		Yes, CM574-RS	Yes, CM574-RS		No, in preparation CM5610-2RS	No, in preparation CM5610-2RS
	203 PROFIBUS DP		Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP		Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP
	208 CAN		Yes, CM588-CN, CM598-CN	Yes, CM588-CN, CM598-CN		Yes, CM598-CN	Yes, CM598-CN
	215 Ethernet TCP/IP		Yes, CM597-ETH	Yes, CM597-ETH		No, in preparation CM5640-2ETH	No, in preparation CM5640-2ETH
	232 EtherCAT		Yes, CM579-ETHCAT	Yes, CM579-ETHCAT		Yes, CM579-ETHCAT	Yes, CM579-ETHCAT
+ 2	236 PROFINET IO RT / PROFISAFE		Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO		Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO
+ 2	242 Safety module SM560-S		Yes, SM560-S-FD-1, SM560-S-FD-4	Yes, SM560-S-FD-1, SM560-S-FD-4		Yes, SM560-S-FD-1, (-FD-4 in preparation)	Yes, SM560-S-FD-1, (-FD-4 in preparation)
+ 2	246 Diagnostic and function		Yes	Yes		Yes	Yes
2	255 Approvals		Yes	Yes		Yes	Yes

3.1.2.1.1 PM572 / PM582

3.1.2.1.2 PM573-ETH / PM583-ETH

A B C	D	E	F	G	H	M	N	0
1 Type			PM573-ETH(-XC)	PM583-ETH(-XC)		PM5630-2ETH(-XC)	PM5650-2ETH(-XC)	PM5670-2ETH(-XC)
2								
3 Supply voltage		Better						
4 24 V DC		Slighty better	Yes	Yes		Yes	Yes	Yes
5 100-240 V AC		Identical / Simila	No	No		No	No	No
6 Type of processor / Processor clock frequency		Lower feature	Freescale ARM Processor 32-bit / 50 MHz	Freescale ARM Processor 32-bit /84 MHz		TI ARM Cortex-A9 32-bit-RISC / 300 MHz	TI ARM Cortex-A9 32-bit-RISC / 600 MHz	TI ARM Cortex-A9 32-bit-RISC / 1 GHz
7 Total RAM memory / Total Flash memory		Not available	32 MB / 16 MB	32 MB / 16 MB		128 MB / 128 MB	256 MB / 512 MB	512 MB / 1024 MB
8 Total user program memory (3)			2048 kB	6144 kB		8 MB	80 MB	160 MB
20 Real-time clock			Yes	Yes		Yes	Yes	Yes
22 Plug-in memory card			Yes	Yes		Yes	Yes	Yes
24 Program execution								
35 Onboard digital inputs			No	No		No	No	No
42 Onboard digital outputs			No	No		No	No	No
56 Onboard digital input /output, configurable channels			No	No		No	No	No
68 Onboard analog outputs			No	No		No	No	No
72 Onboard analog inputs			No	No		-	-	-
76 Max. number of centralized inputs/outputs								
85 Max. number of decentralized inputs/outputs			depends on the used standard fieldbus	depends on the used standard fieldbus		depends on the used standard fieldbus	depends on the used standard fieldbus	depends on the used standard fieldb
88 Option board slots for extension			No	No		No	No	No
99 Internal interfaces								
100 COM1			Yes	Yes		Yes	Yes	Yes
110 COM2			Yes	Yes		No	No	No
119 FieldbusPlug			Yes, but FieldBusPlug is out phased	Yes, but FieldBusPlug is out phased		No	No	No
126 CAN Interface			No	No		Yes	Yes	Yes
136 Ethernet			Yes	Yes		Yes	Yes	Yes
185 Terminal base for connection / CMS			Yes / No	Yes / No		Yes / No	Yes / No	Yes / No
195 Supported communication with Communication Mod	ule							
196 Serial communication based			Yes, CM574-RS	Yes, CM574-RS		No, in preparation CM5610-2RS	No, in preparation CM5610-2RS	No, in preparation CM5610-2RS
203 PROFIBUS DP			Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP		Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP
208 CAN			Yes, CM588-CN, CM598-CN	Yes, CM588-CN, CM598-CN		Yes, CM598-CN	Yes, CM598-CN	Yes, CM598-CN
215 Ethernet TCP/IP			Yes, CM597-ETH	Yes, CM597-ETH		No, in preparation CM5640-2ETH	No, in preparation CM5640-2ETH	No, in preparation CM5640-2ETH
232 EtherCAT			Yes, CM579-ETHCAT	Yes, CM579-ETHCAT		Yes, CM579-ETHCAT	Yes, CM579-ETHCAT	Yes, CM579-ETHCAT
236 PROFINET IO RT / PROFISAFE			Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO		Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO
242 Safety module SM560-S			Yes, SM560-S-FD-1, SM560-S-FD-4	Yes, SM560-S-FD-1, SM560-S-FD-4		Yes, SM560-S-FD-1, (-FD-4 in preparation	(es, SM560-S-FD-1, (-FD-4 in preparation	Yes, SM560-S-FD-1, (-FD-4 in preparati
246 Diagnostic and function			Yes	Yes		Yes	Yes	Yes
255 Approvals			Yes	Yes		Yes	Yes	Yes

3.1.2.1.3 PM585-ETH

12	A B C I	DE	F	G	L	М	N
1	1 Type		PM585-ETH		PM5630-2ETH(-XC)	PM5650-2ETH(-XC)	PM5670-2ETH(-XC)
2	2						
3	3 Supply voltage	Better					
4	4 24 V DC	Slighty better	Yes		Yes	Yes	Yes
5	5 100-240 V AC	Identical / Similar	No		No	No	No
6	6 Type of processor / Processor clock frequency	Lower feature	Freescale ARM Processor 32-bit /400 MHz		TI ARM Cortex-A9 32-bit-RISC / 300 MHz	TI ARM Cortex-A9 32-bit-RISC / 600 MHz	TI ARM Cortex-A9 32-bit-RISC / 1 GHz
7	7 Total RAM memory / Total Flash memory	Not available	64 MB / 32 MB		128 MB / 128 MB	256 MB / 512 MB	512 MB / 1024 MB
+ 8	8 Total user program memory (3)		7680 kB		8 MB	80 MB	160 MB
+ 20	20 Real-time clock		Yes		Yes	Yes	Yes
+ 22	22 Plug-in memory card		Yes		Yes	Yes	Yes
+ 24	24 Program execution						
+ 35	35 Onboard digital inputs		No		No	No	No
+ 42	42 Onboard digital outputs		No		No	No	No
+ 56	56 Onboard digital input /output, configurable channels		No		No	No	No
	58 Onboard analog outputs		No		No	No	No
	72 Onboard analog inputs		No		-	-	-
+ 76	76 Max. number of centralized inputs/outputs						
	35 Max. number of decentralized inputs/outputs		depends on the used standard fieldbus		depends on the used standard fieldbus	depends on the used standard fieldbus	depends on the used standard fieldbus
+ 88	38 Option board slots for extension		No		No	No	No
	99 Internal interfaces						
	00 COM1		Yes		Yes	Yes	Yes
	10 COM2		Yes		No	No	No
	19 FieldbusPlug		Yes, but FieldBusPlug is out phased		No	No	No
+ 12	26 CAN Interface		No		Yes	Yes	Yes
+ 13	36 Ethernet		Yes		Yes	Yes	Yes
	85 Terminal base for connection / CMS		Yes / No		Yes / No	Yes / No	Yes / No
	95 Supported communication with Communication Module						
	96 Serial communication based		Yes, CM574-RS		No, in preparation CM5610-2RS	No, in preparation CM5610-2RS	No, in preparation CM5610-2RS
	03 PROFIBUS DP		Yes, CM582-DP, CM592-DP		Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP
	08 CAN		Yes, CM588-CN, CM598-CN		Yes, CM598-CN	Yes, CM598-CN	Yes, CM598-CN
	15 Ethernet TCP/IP		Yes, CM597-ETH		No, in preparation CM5640-2ETH	No, in preparation CM5640-2ETH	No, in preparation CM5640-2ETH
	32 EtherCAT		Yes, CM579-ETHCAT		Yes, CM579-ETHCAT	Yes, CM579-ETHCAT	Yes, CM579-ETHCAT
	36 PROFINET IO RT / PROFISAFE		Yes, CM589-PNIO, CM579-PNIO		Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO
	42 Safety module SM560-S		Yes, SM560-S-FD-1, SM560-S-FD-4		Yes, SM560-S-FD-1, (-FD-4 in preparation	(es, SM560-S-FD-1, (-FD-4 in preparation	es, SM560-S-FD-1, (-FD-4 in preparation)
	46 Diagnostic and function		Yes		Yes	Yes	Yes
25	55 Approvals		Yes		Yes	Yes	Yes

3.1.2.1.4 PM590-ETH / PM591-ETH

1 2 A B C	DE	F	G	H N	0	р
1 Type		PM590-ETH	PM591-ETH(-XC)	PM5650-2ETH(-XC)	PM5670-2ETH(-XC)	PM5675-2ETH(-XC)
2						
3 Supply voltage	Better					
4 24 V DC	Slighty better	Yes	Yes	Yes	Yes	Yes
5 100-240 V AC	Identical / Similar	No	No	No	No	No
6 Type of processor / Processor clock frequency	Lower feature	Freescale ARM Processor 32-bit / 400 MHz	Freescale ARM Processor 32-bit /400 MHz	TI ARM Cortex-A9 32-bit-RISC / 600 MHz	TI ARM Cortex-A9 32-bit-RISC / 1 GHz	TI ARM Cortex-A9 32-bit-RISC / 1 GHz
7 Total RAM memory / Total Flash memory	Not available	64 MB / 32 MB	64 MB / 32 MB	256 MB / 512 MB	512 MB / 1024 MB	512 MB / 1024 MB
+ 8 Total user program memory (3)		13316 kB	17924 kB	80 MB	160 MB	160 MB
+ 20 Real-time clock		Yes	Yes	Yes	Yes	Yes
+ 22 Plug-in memory card		Yes	Yes	Yes	Yes	Yes
+ 24 Program execution						
+ 35 Onboard digital inputs		No	No	No	No	No
+ 42 Onboard digital outputs		No	No	No	No	No
+ 56 Onboard digital input /output, configurable channels		No	No	No	No	No
+ 68 Onboard analog outputs		No	No	No	No	No
+ 72 Onboard analog inputs		No	No	-	-	-
+ 76 Max. number of centralized inputs/outputs						
+ 85 Max. number of decentralized inputs/outputs		depends on the used standard fieldbus	depends on the used standard fieldbus		depends on the used standard fieldbus	depends on the used standard fieldbus
+ 88 Option board slots for extension		No	No	No	No	No
99 Internal interfaces						
+ 100 COM1		Yes	Yes	Yes	Yes	Yes
+ 110 COM2		Yes	Yes	No	No	No
+ 119 FieldbusPlug		Yes, but FieldBusPlug is out phased	Yes, but FieldBusPlug is out phased	No	No	No
+ 126 CAN Interface		No	No	Yes	Yes	Yes
+ 136 Ethernet		Yes	Yes	Yes	Yes	Yes
+ 185 Terminal base for connection / CMS		Yes / No	Yes / No	Yes / No	Yes / No	Yes / No
195 Supported communication with Communication Modu	e					
+ 196 Serial communication based		Yes, CM574-RS	Yes, CM574-RS	No, in preparation CM5610-2RS	No, in preparation CM5610-2RS	No, in preparation CM5610-2RS
+ 203 PROFIBUS DP		Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP	Yes, CM582-DP, CM592-DP
+ 208 CAN		Yes, CM588-CN, CM598-CN	Yes, CM588-CN, CM598-CN	Yes, CM598-CN	Yes, CM598-CN	Yes, CM598-CN
+ 215 Ethernet TCP/IP		Yes, CM597-ETH	Yes, CM597-ETH	No, in preparation CM5640-2ETH	No, in preparation CM5640-2ETH	No, in preparation CM5640-2ETH
+ 232 EtherCAT		Yes, CM579-ETHCAT	Yes, CM579-ETHCAT	Yes, CM579-ETHCAT	Yes, CM579-ETHCAT	Yes, CM579-ETHCAT
+ 236 PROFINET IO RT / PROFISAFE		Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO	Yes, CM589-PNIO, CM579-PNIO
+ 242 Safety module SM560-S		Yes, SM560-S-FD-1, SM560-S-FD-4	Yes, SM560-S-FD-1, SM560-S-FD-4	Yes, SM560-S-FD-1, (-FD-4 in preparation		
+ 246 Diagnostic and function		Yes	Yes	Yes	Yes	Yes
255 Approvals		Yes	Yes	Yes	Yes	Yes

3.1.2.2 Overview feature list of AC500-eCo V2 with potential AC500-eCo V3 substitute PLCs

3.1.2.2.1 PM554-xP

12	A	В	C	D	E	F	G	н	М	N	0	P
1	Туре					PM554-TP	PM554-RP(-AC)		PM5012-T-ETH	PM5012-R-ETH	PM5032-T-ETH	PM5032-R-ETH
2	Supply voltag	e			Better							
4		24 V DC			Slighty better	Yes	Yes		Yes	Yes	Yes	Yes
5		100-240 V A	C		Identical / Similar	No	Yes		No	No	No	No
6	Type of proce	ssor / Proces	sor clock frequency		Lower feature	Freescale ARM Proce	ssor 32-bit / 50 MHz		TI ARM Cortex-A9 3	2-bit-RISC / 300 MHz	TI ARM Cortex-A9 3	2-bit-RISC / 300 MHz
7	Total RAM m	emory / Total I	Flash memory		Not available	16 MB	/ 4 MB		128 MB	/ 128 MB	128 MB	/ 128 MB
+ 8	Total user pr	ogram memor	y (3)			142	kB		1	MB	2	MB
+ 20	Real-time clo	ck				Optional with TA561	RTC or TA562-RS-RTC		Optional, use optio	n board TA5131-RTC	1	'es
+ 22	Plug-in mem	ory card				Optional with MC	503 option board		Yes, onboard micro	memory card socket	Yes, onboard micro	memory card socket
+ 24	Program exe	cution										
+ 35	Onboard digi	tal inputs				Ye	25		Y	es	<u> </u>	'es
	Onboard digi					Ye	15		Y	es	۱ ۱	'es
+ 56	Onboard digi	tal input /out	out, configurable channels			N				ło		'es
+						No, specific	CPU version			ossible using additional option		ossible using additional option
	Onboard ana								board TA5	126-2AO-UI	board TA5	126-2AO-UI
+ 72	Onboard ana	<u> </u>				No, specific	CPU version		With opt. boards TA5120-2AI-U	I, TA5122-2AI-TC, TA5123-2AI-RTD	With opt. boards TA5120-2AI-U	I, TA5122-2AI-TC, TA5123-2AI-RTD
+ 76			l inputs/outputs			Little bit more a			No local I/O extension e	xcept with option boards		/ Less analog
			zed inputs/outputs			depends on the used				standard fieldbus (1)		standard fieldbus (1)
	Option board		nsion			2 Dedica	ted slots			e of existing option boards, same o		ral slots per CPU
	Internal inter	faces								y using additional option boards f		
	0 COM1					Ye			Optional with	option boards		option boards
	0 COM2					Optional with				ło		option boards
	9 сомз					N				10		No.
	8 Ethernet					No, but with sp				es		'es
	6 Diagnostic an	d function				Ye	*5			es	1	'es
	1 Approvals							See d	etailled page 272 or www.abb.co	m/plc		
182												
						TA562-RS-RTC, TA562-RS or new TA	569-RS-ISO.					
184	4 (3) Total user	program men	nory: contains user progra	im code, d	ata and web server							

CONVERTING AN AC500 V2 PROJECT TO AN AC500 V3 PROJECT

3.1.2.2.2 PM564-xP

A B C	DE	F	G	н	M	N	0	Р
1 Type		PM564-TP	PM564-RP(-AC)		PM5012-T-ETH	PM5012-R-ETH	PM5032-T-ETH	PM5032-R-ETH
2								
3 Supply voltage	Better							
4 24 V DC	Slighty better	Yes	Yes		Yes	Yes	Yes	Yes
5 100-240 V AC	Identical / Similar	No	Yes		No	No	No	No
6 Type of processor / Processor clock frequency	Lower feature	Freescale ARM Proc	essor 32-bit / 50 MHz		TI ARM Cortex-A9 3	2-bit-RISC / 300 MHz	TI ARM Cortex-A9 32	-bit-RISC / 300 MHz
7 Total RAM memory / Total Flash memory	Not available	16 ME	3 / 4 MB		128 MB	/ 128 MB	128 MB /	128 MB
8 Total user program memory (3)		14	2 kB		1	MB	2 M	//B
20 Real-time clock		Optional with TA561	I-RTC or TA562-RS-RTC		Optional, use optio	n board TA5131-RTC	Ye	15
22 Plug-in memory card		Optional with M	C503 option board		Yes, onboard micro	memory card socket	Yes, onboard micro	memory card socket
24 Program execution								
35 Onboard digital inputs		<u> </u>	'es		Y	es	Ye	5
42 Onboard digital outputs		<u>۱</u>	/es		Y	es	Ye	s
56 Onboard digital input /output, configurable channels			No			lo	Ye	
68 Onboard analog outputs		Y	'es		No onboard analog I/O, but po		No onboard analog I/O, but po	
72 Onboard analog inputs			'es			, TA5122-2AI-TC, TA5123-2AI-RTD	With opt. boards TA5120-2AI-UI,	
76 Max. number of centralized inputs/outputs			analog possible		· · · · · · · · · · · · · · · · · · ·	xcept with option boards	More digital ,	
85 Max. number of decentralized inputs/outputs			standard fieldbus (1)			standard fieldbus (1)	depends on the used s	
88 Option board slots for extension		2 Dedic	ated slots				ption board can be used on sever	al slots per CPU
99 Internal interfaces					No onboard serial interface, only			
100 COM1			'es		Optional with		Optional with	
110 COM2		Optional with	option boards		Ν	lo	Optional with	option boards
119 COM3			No			lo	N	_
128 Ethernet		No, but with s	pecial versions		Y	es	Ye	
176 Diagnostic and function		Y	'es		1	es	Ye	s
181 Approvals				See o	detailled page 272 or www.abb.co	m/plc		
182								
183 (1) Real-time clock requires optional TA561-RTC or TA56		or new TA569-RS-ISO.						
184 (3) Total user program memory: contains user program	code, data and web server							

3.1.2.2.3 PM554-xP-ETH

1 2 🖌 A B C	DE	F	G	L	M	N	0
1 Type		PM554-TP-ETH		PM5032-T-ETH	PM5032-R-ETH	PM5052-T-ETH	PM5052-R-ETH
2							
3 Supply voltage	Better						
4 24 V DC	Slighty better	Yes		Yes	Yes	Yes	Yes
5 100-240 V AC	Identical / Similar	No		No	No	No	No
6 Type of processor / Processor clock frequency	Lower feature	Freescale ARM Processor 32-bit / 50 MHz		TI ARM Cortex-A9 3	2-bit-RISC / 300 MHz	TI ARM Cortex-A9 3	2-bit-RISC / 300 MHz
7 Total RAM memory / Total Flash memory	Not available	16 MB / 4 MB			/ 128 MB	128 MB	/ 128 MB
+ 8 Total user program memory (3)		654 kB		2	MB	4	ИВ
+ 20 Real-time clock		Optional with TA561-RTC or TA562-RS-RTC		١	/es	Y	25
+ 22 Plug-in memory card		Optional with MC503 option board		Yes, onboard micro	memory card socket	Yes, onboard micro	memory card socket
+ 24 Program execution							
+ 35 Onboard digital inputs		Yes		١	(es	Y	25
+ 42 Onboard digital outputs		Yes		١	/es	Y	25
+ 56 Onboard digital input /output, configurable channels		No			/es		es
+ 68 Onboard analog outputs		No, specific CPU version			ossible using additional option		ssible using additional option
+ 72 Onboard analog inputs		No, specific CPU version			I, TA5122-2AI-TC, TA5123-2AI-RTD		, TA5122-2AI-TC, TA5123-2AI-RTD
+ 76 Max. number of centralized inputs/outputs		Little bit more analog possible			/ Less analog		More analog
+ 85 Max. number of decentralized inputs/outputs		depends on the used standard fieldbus (1)			l standard fieldbus (1)		standard fieldbus (1)
+ 88 Option board slots for extension		2 Dedicated slots		Each slot can be used for all typ	e of existing option boards, same o	ption board can be used on seve	al slots per CPU
99 Internal interfaces							
+ 100 COM1		Yes			option boards		option boards
+ 110 COM2		Optional with option boards		Optional with	n option boards		option boards
+ 119 COM3 + 128 Ethernet		No			No	Optional with	option boards
		Yes			/es		
+ 176 Diagnostic and function		Yes			/es	Y	es
181 Approvals							
182							
183 (1) Real-time clock requires optional TA561-RTC or TA562-R		ew TA569-RS-ISO.					
184 (3) Total user program memory: contains user program cod	e, data and web server						

3.1.2.2.4 PM564-xP-ETH

A B C	DE	F	G	Н	М	N	0	Р
1 Type		PM564-TP-ETH	PM564-RP-ETH(-AC)		PM5032-T-ETH	PM5032-R-ETH	PM5052-T-ETH	PM5052-R-ETH
2								
3 Supply voltage	Better							
4 24 V DC	Slighty better	Yes	Yes		Yes	Yes	Yes	Yes
5 100-240 V AC	Identical / Similar	No	Yes		No	No	No	No
6 Type of processor / Processor clock frequency	Lower feature	Freescale ARM Proce	essor 32-bit / 50 MHz		TI ARM Cortex-A9 32	-bit-RISC / 300 MHz	TI ARM Cortex-A9 3	32-bit-RISC / 300 MHz
7 Total RAM memory / Total Flash memory	Not available	16 MB	/ 4 MB		128 MB /	128 MB	128 ME	/ 128 MB
8 Total user program memory (3)		654	4 kB		2 N	ИB	4	MB
20 Real-time clock		Optional with TA561	-RTC or TA562-RS-RTC		Ye	15		Yes
22 Plug-in memory card		Optional with MC	503 option board		Yes, onboard micro r	memory card socket	Yes, onboard micro	o memory card socket
24 Program execution								
35 Onboard digital inputs		Y	es		Ye	15		Yes
42 Onboard digital outputs		Y	es		Ye	15		Yes
56 Onboard digital input /output, configurable channels		N	lo		Ye	25		Yes
68 Onboard analog outputs		Y	es		No onboard analog I/O, but pos	ssible using additional option	No onboard analog I/O, but p	ossible using additional op
72 Onboard analog inputs		Y	es		With opt. boards TA5120-2AI-UI,	TA5122-2AI-TC, TA5123-2AI-RTD	With opt. boards TA5120-2AI-U	JI, TA5122-2AI-TC, TA5123-2AI
76 Max. number of centralized inputs/outputs		Little bit more a	analog possible		More digital /	/ Less analog	More digital	/ More analog
85 Max. number of decentralized inputs/outputs		depends on the used	standard fieldbus (1)		depends on the used s	standard fieldbus (1)	depends on the used	d standard fieldbus (1)
88 Option board slots for extension		2 Dedica	ited slots		Each slot can be used for all type	of existing option boards, same	option board can be used on seve	eral slots per CPU
99 Internal interfaces								
100 COM1		Y	es		Optional with o	option boards	Optional wit	h option boards
110 COM2		Optional with	option boards		Optional with o	option boards	Optional wit	h option boards
119 COM3		N	lo		N	0	Optional wit	h option boards
128 Ethernet		Y	es		Ye	15		
176 Diagnostic and function		Y	es		Ye	is		Yes
181 Approvals								
182								
183 (1) Real-time clock requires optional TA561-RTC or TA562	2-RS-RTC. (2) COM2 requires TA562-RS-RTC, TA562-RS	or new TA569-RS-ISO.						
184 (3) Total user program memory: contains user program of	ode, data and web server							

3.1.2.2.5 PM556-TP-ETH

1 2 A B C	DE	F	G	L	М	N	0
1 Type		PM556-TP-ETH		PM5052-T-ETH	PM5052-R-ETH	PM5072-T-2ETH	PM5072-T-2ETHW(2)
2							
3 Supply voltage	Better						
4 24 V DC	Slighty better	Yes		Yes	Yes	Yes	Yes
5 100-240 V AC	Identical / Similar	No		No	No	No	No
6 Type of processor / Processor clock frequency	Lower feature	Freescale ARM Processor 32-bit / 50 MHz		TI ARM Cortex-A9 3	32-bit-RISC / 300 MHz	TI ARM Cortex-A9	32-bit-RISC / 300 MHz
7 Total RAM memory / Total Flash memory	Not available	16 MB / 4 MB		128 ME	3 / 128 MB	128 MB	8 / 128 MB
8 Total user program memory (3)		1666 kB		4	MB	8	MB
20 Real-time clock		Optional with TA561-RTC or TA562-RS-RTC			Yes		Yes
22 Plug-in memory card		Optional with MC503 option board		Yes, onboard micro	o memory card socket	Yes, onboard micro	o memory card socket
24 Program execution							
 35 Onboard digital inputs 		Yes			Yes		Yes
42 Onboard digital outputs		Yes			Yes		Yes
56 Onboard digital input /output, configurable channels		No			Yes		Yes
68 Onboard analog outputs		No, specific CPU version		No onboard analog I/O, but p	ossible using additional option		ossible using additional option
72 Onboard analog inputs		No, specific CPU version		With opt. boards TA5120-2AI-L	JI, TA5122-2AI-TC, TA5123-2AI-RTD	With opt. boards TA5120-2AI-U	JI, TA5122-2AI-TC, TA5123-2AI-RTD
76 Max. number of centralized inputs/outputs		Little bit more analog possible		More digital	/ More analog	More digita	/ More analog
85 Max. number of decentralized inputs/outputs		depends on the used standard fieldbus (1)			d standard fieldbus (1)		d standard fieldbus (1)
88 Option board slots for extension		2 Dedicated slots		Each slot can be used for all typ	e of existing option boards, same (option board can be used on seve	ral slots per CPU
99 Internal interfaces							
100 COM1		Yes		Optional wit	h option boards	Optional wit	h option boards
110 COM2		Optional with option boards		Optional wit	h option boards	Optional wit	h option boards
119 COM3		No		Optional wit	h option boards	Optional wit	h option boards
128 Ethernet		Yes			Yes		Yes
176 Diagnostic and function		Yes			Yes		Yes
181 Approvals							
182							
183 (1) Real-time clock requires optional TA561-RTC or TA56	62-RS-RTC. (2) COM2 requires TA562-RS-RT	C, TA562-RS or new TA569-RS-ISO.					
184 (3) Total user program memory: contains user program	code, data and web server						

3.1.2.2.6 PM566-TP-ETH

1 2	A	В	C	D		E	F	G	L	M	N	0
	1 Type						PM566-TP-ETH		PM5052-T-ETH	PM5052-R-ETH	PM5072-T-2ETH	PM5072-T-2ETHW(2)
	2											
	3 Supply voltage	2			Better							
	4	24 V DC			Slighty better		Yes		Yes	Yes	Yes	Yes
	5	100-240 V AC			Identical / Simi	ar	No		No	No	No	No
	6 Type of proces	sor / Processo	or clock frequency		Lower feature		Freescale ARM Processor 32-bit / 50 MHz		TI ARM Cortex-A9 3	2-bit-RISC / 300 MHz	TI ARM Cortex-A9 3	2-bit-RISC / 300 MHz
	7 Total RAM me	mory / Total Fl	ash memory		Not available		16 MB / 4 MB		128 MB	/ 128 MB	128 MB	/ 128 MB
	8 Total user pro		(3)				1666 kB		4	MB	8	MB
+ 2	20 Real-time cloc	k					Optional with TA561-RTC or TA562-RS-RTC		١	/es	<u>ر</u>	'es
+ 2	22 Plug-in memo	ry card					Optional with MC503 option board		Yes, onboard micro	memory card socket	Yes, onboard micro	memory card socket
+ 2	24 Program exec	ution										
+ 3	35 Onboard digita	al inputs					Yes			(es	·	'es
	42 Onboard digita						Yes		<u>۱</u>	/es	<u> </u>	'es
+ :	56 Onboard digit	al input /outpu	it, configurable channe	ls			No		<u> </u>	/es	<u> </u>	'es
	58 Onboard analo						Yes		No onboard analog I/O, but p	ossible using additional option	No onboard analog I/O, but p	ossible using additional option
+ 7	72 Onboard analo	og inputs					Yes		With opt. boards TA5120-2AI-U	I, TA5122-2AI-TC, TA5123-2AI-RTD	With opt. boards TA5120-2AI-U	I, TA5122-2AI-TC, TA5123-2AI-RTD
	76 Max. number						Little bit more analog possible			/ More analog		/ More analog
	85 Max. number						depends on the used standard fieldbus (1)		depends on the used	l standard fieldbus (1)	depends on the used	standard fieldbus (1)
+ 8	88 Option board	slots for exten	sion				2 Dedicated slots		Each slot can be used for all type	e of existing option boards, same o	ption board can be used on seve	ral slots per CPU
	99 Internal interf	aces										
+ 1	00 COM1						Yes		Optional with	option boards	Optional with	option boards
+ 1	10 COM2						Optional with option boards		Optional with	option boards	Optional with	option boards
+ 1	19 COM3						No		Optional with	option boards	Optional with	option boards
+ 1	28 Ethernet						Yes		۱ ۱	(es	ر بر	'es
+ 1	76 Diagnostic and	function					Yes			(es	۱	'es
1	81 Approvals											
1	82											
1	83 (1) Real-time o	lock requires	optional TA561-RTC or	TA562-RS-RTC.	(2) COM2 requires	TA562-RS-RTC, TA562-RS or	new TA569-RS-ISO.					
1	84 (3) Total user	program memo	ory: contains user prog	gram code, data	a and web server							



3.1.3 Safety CPUs

AC500 V2	AC500 V3
On all coupler slots:	On all coupler slots:
SM560-S,	SM560-S
SM560-S-FD1,	
SM560-S-FD4	
	On all coupler slots: SM560-S, SM560-S-FD1,

3.1.4 AC31-Replacement

Device	AC500 V2	AC500 V3
AC31-Replacement CPUs	07КТ98-хх-уу	
AC31 Replacement CS31-IOs	07AC9x-yy, 07Al91-AD, 07DC9x- AD	

3.1.5 Hardware Interfaces

Fieldbus / Interface	AC500 V2	AC500 V3	AC500-eCo V3
Serial communication	COM1(terminal block), - RS232, RS422, RS485	COM1(terminal block)	Option boards (termi- nal block)
* protocols see chap-	, ,	CAN terminal block	- RS232, RS232 isolated
ter below	COM2(D-SUB) - RS232, RS422, RS485		- RS485, RS485 iso- lated
Ethernet interfaces * protocols see chap- ter below	ETH1 and coupler CM597-ETH PM591-2ETH additional ETH2,	ETH1 and ETH2	ETH1 PM5072 additional ETH2
	PM595-4ETH additional ETH2, ETH3, ETH4,		

3.1.6 Fieldbus communication modules (CM)

Fieldbus / Protocol	AC500 V2	AC500 V3	AC500-eCo V3
Profinet Controller	CM579-PNIO with PM595-4ETH op- tional on board of ETH3 and ETH4.	CM579-PNIO	Not available
Profinet Device	CM589-PNIO and CM589-PNIO-4	CM589-PNIO and CM589-PNIO-4	Not available
Profibus Master	CM592-DP	CM592-DP	Not available
Profibus Slave	CM582-DP	CM582-DP	Not available

CANopen Manager (Master)	CM578-CN CM598-CN	Onboard (terminal block) and CM598-CN	Not available
CANopen Device (Slave)	CM588-CN	Not available	Not available
CAN 2A/2B	СМ598-СN	Onboard (terminal block) and CM598-CN	Not available
EtherCAT	CM579-ETHCAT with PM595-4ETH op- tional on board of ETH3 and ETH4.	CM579-ETHCAT	Not yet available
Modbus TCP	СМ577-ЕТН СМ597-ЕТН	Only onboard Interface	Only onboard Interface

3.1.7 Communication Interface modules

Fieldbus / Protocol	AC500 V2	AC500 V3	АС500-еСо V3
Profinet	CI501-PNIO CI502-PNIO CI504-PNIO CI506-PNIO	CI501-PNIO CI502-PNIO	n.a.
Profibus Master	CI541-DP CI542-DP	CI541-DP CI542-DP	n.a.
CANopen	CI581 CI582	CI581 * CI582 * *) only to onboard CANopen Manager	
EtherCAT	CI511-ETHCAT CI512-ETHCAT	CI511-ETHCAT CI512-ETHCAT	
Modbus TCP	CI521-MODTCP CI522-MODTCP	CI521-MODTCP CI522-MODTCP	CI521-MODTCP CI522-MODTCP

3.1.8 Other devices

Module(s) / Device(s)	AC500 V2	AC500 V3	AC500-eCo V3
IO-Modules	All	All	All
Function Modules	DC541-CM	Not supported	Not available
	FM562-PTO	Not supported	Use on board PTO out- puts

3.1.9 Memory Card

	AC500 V2	AC500 V3	
Size	Max. 2 GB (ABB tested Cards)	Max. 8 GB from ABB	
Format	Standard	SDHC supported, SDXC is not supported	
File System	DOS 8.3 – Filesystem (Short Names)	FAT 32, Long Names	
Card Func- tion	Function Description of Card in Init File SDCARD.INI	Function Description of Card in Init File SDCARD.INI	
Functional- ity	 One Card for FW Update (CPU, CM Module, Display) User Data Retain-/Persistent- Data Project Source Code Application Prog. CM Module Conf. 	 One Card for FW Update (System, Boot, Update, Display) User Data Retain-/Persistent- Data Project Source Code Application Prog. Trends and Alarms Backup and Restore 	

Hint: For AC500 V3 the memory card is optional. All functionalities can also be performed via Automation Builder or the internal userdisk memory.

For AC500 V3 there is no SD Card mandatory. All Functionalities can also be performed via Automation Builder or the internal userdisk memory in the AC500.

3.2 Protocols

(hij)

3.2.1 Serial protocols

Fieldbus / Protocol	AC500 V2	AC500 V3	AC500-eCo V3
Modbus RTU Client and Sever	COM1 and COM2 Coupler: CM574-RS	Onboard COM1	Optional Up to 3 option boards available (terminal block)
CS31	COM1 CM574-RS	Not yet available	Not yet available
RCOM	CM574-RCOM	Not available	Not available
ASCII	COM1 and COM2	Not yet available	Not yet available
MULTI	COM1 and COM2	Not yet available	Not yet available

SysLibCom – free prog	COM1 and COM2	CAA SerialCom	CAA SerialCom
BACNET MS/TP	COM1 and COM2	COM1 and COM2	Option board TA5142-RS485I- BacNet
CANopen Manager (Master)	СМ598-СN	Onboard (terminal block) and CM598-CN	Not available
CANopen Device (Slave)	СМ588-СN	Not available	Not available
CAN 2A/2B	СМ598-СN	Onboard (terminal block) and CM598-CN	Not available

3.2.2 Ethernet based protocols

AC500 V2	AC500 V3
Modbus TCP, Client + Server	Modbus TCP, Client + Server
TCP/IP Sockets	TCP/IP Sockets (CAA)
UDP (Standard UDP + ABB UDP)	TCP/UDP
Online access (configurable by the user, 3S and ABB drivers)	3S Online access (configurable only regarding security settings)
IEC60870-5-104 (Control - & Substation)	IEC60870-5-104 (Control - & Substation) Control station depending on PLC type
	IEC61850 (MMS Server, GOOSE, Edition 1)
SNTP (Client + Server)	SNTP, NTP (Client + Server)
SMTP	SMTP (including encryption)
SNMP	SNMP
FTP Server (Client as Lib)	FTP(s) Server (connected to both Interfaces)
WEBserver	CODESYS HTML5 WEB Visualization
	CODESYS network variables
OPC DA (Server installed at PC)	OPC DA (Server installed at PC)
	OPC UA Server
	OPC UA Client
MQTT	MQTT
НТТР	HTTP + HTTPS
	Client Protocols added to Protocols
	Server Protocols added below dedicated Ether- net Port
BACnet B-ASC	BACnet B-BC

3.2.3 SMTP protocol

The biggest difference between V2 and V3 is, that v2 does not support TLS or SSL encryption. As all mail provider use encryption today, this functionality is only usable for local mail sending using a local unencrypted mail software.

The second differnce is that in V3 the functionality is done in function blocks which must be prepared properly. In V2 there is a mask to put in everything. This is easier to use but on the contrary the possibilities are more limited than a function bock based programming.

The V2 SMTP has a mask to put in the parameters:

SMTP configuration	1		
Server	213 . 165 . 64 . 21		
Port	25		
Mail address	mail@gmx.de		
User name	username@gmx.de		
Password	•••••		
Domain	gmx.de		
Sender name	mail		
Authentication	scheme AUTH LOGIN 🗸 🗸		

In V3, function blocks are used:

(*PREPARE MAIL*)		
BodyToSend	:= MailContent;	<pre>// Content of the mail</pre>
mailToSend.sSubject	:= MailSubject;	<pre>// Subject of the mail</pre>
mailToSend.sMailTo	:= MailTo;	<pre>// Destination mail address</pre>
mailToSend.sMailFrom	:= MailFrom;	
mailToSend.sMailCC	:= ''';	
mailToSend.sMailBCC	:= ''';	
userInfo.sUserName:=Setting	gs.UserName;	
userInfo.sPassword:=Setting	gs.Password;	
mailToSend.Body.pString	:= ADR (BodyToSend);	
mailToSend.Body.uiSize	:= INT_TO_UINT (LEN (Body	yToSend));

3.3 Libraries

The following tables can be taken as rough orientation where to find the required functions and function blocks. Please consider:

- In some cases, the AC500 V3 do not offer the complete functionality as the AC500 V2 library. Typically, mostly used functions and function blocks are available so that a project conversion can be done.
- There is not always a one-to-one relation between AC500 V2 and AC500 V3 libraries.
- They are not complete and might have changed since last release of this document. New libraries for AC500 V3 are continuously developed.
- Final decision, if all required functionality is available with AC500 V3 has to be made on function / function block level. The full documentation of all libraries that are installed together with Automation Builder is available online from the internet. Latest version is the Automation Builder 2.5 library documentation.

3.3.1 AC500 System libraries

Торіс	AC500 V2	AC500 V3	Remark
	SysExt_AC500_V10.lib, SysInt_AC500_V10.lib, SysIntExt_AC500_V13.lib	Pm, IO	Processor Module and IO Util- ities
AC500	BusDiag.lib, Diag_AC500_V20.lib	Diag, DiagCpu, DiagUtil, DiagIoBus, DiagTypes, DiagBase, DiagHistory, DiagS500	Diagnosis libraries
	CD522_AC500_V13.lib	CD522	supporting functions of en- coder & PWM module CD522
	Counter_AC500_V20.lib	10	
	RTC_AC500_V20.lib	Pm	
	FBP.lib	n.a.	
	DC541_AC500_V11.lib	n.a.	
	Serie90_AC500_V10.lib	n.a.	
AC500 (Eco)	n.a.	EcoUtils	utility functions for AC500 ECO PLCs
	OnBoardIO_AC500_V13.lib	OnboardIO	

3.3.2 Communication libraries

Торіс	AC500 V2	AC500 V3	Remark
FTP client	FTPClient_AC500_V22.lib	Planned for Q3 2023	
BACnet	BACnet_BASC_AC500_V28.lib	BACnet	V3 has B-BC profile plus AB plugin (V2 only B-ASC li- brary)
IEC 61850	n.a.	IEC61850Server	
IEC 60870	IEC60870_AC500_V20.lib	IEC60870_5_104	
	EtherCAT_AC500_V13.lib, EtherCAT_CS_AC500.lib	EtherCAT, EcatBase	
EtherCAT			
	CAMSWITCH_AC500_V13.lib, MCX_AC500_Vx.lib		
CAN	CANopen_AC500_V11.lib	CM598Can	CM598-CAN specific function blocks
Serial	ASCII_AC500_V10.lib	Com	Serial Communica- tion for AC500 V3
Modbus RTU	Modbus_AC500_V10.lib, MODBUS_Ext_AC500_V20.lib	ModbusRtu	
Modbus Ethernet	Ethernet_AC500_V10.lib	ModbusTcp	
Com. Profibus	PROFIBUS_AC500_V10.lib	CM592Profibus	
Com. Profinet	PROFINET_AC500_V13.lib, PROFINET_Ext_AC500_V20.lib	PnioCntrl	
	MqttClient_AC500_V28.lib	Mqtt	
Com. MQTT	JSON_AC500_V28.lib	Json	
Com. Ethernet	Ethernet_AC500_V10.lib	Ethernet	Ethernet communi- cation with AC500 devices.
CS31	CS31_AC500_V20.lib	n.a.	
Arcnet	ARCNET_AC500_V12.lib, ARCNETExt_AC500_V12.lib	n.a.	
Rcom	RCOM_AC500_V13.lib	n.a.	
DeviceNet	DeviceNet_AC500_V11.lib	n.a.	

3.3.3 Safety libraries

The Safety libraries run exclusively inside the Safety PLC which is independent of the used AC500 PLC (AC500 V2 or AC500 V3).

3.3.4 Application / Product libraries

Торіс	AC500 V2	AC500 V3	Remark
	MC_Base_AC500_V11.lib, MC_Blocks_AC500_V11.lib, CompactMotionControl_AC500_V12.lib	MotionControl	
	ECAT_AC500_APPL_V21.lib	Ecat_CiA402	
	MathFunctions_AC500_V23.lib	MathFunctions	
Motion Eco	n.a.	MotionContro- IEco	
Motion PTO	PTO_FM562_MC_support_Vxx.lib	MotionContro- IEco	FM562 is not supported in V3. Use onboard PTO of AC500- eCo V3.
Motion Load control	n.a.	MotionControl- Load	
Motion coordi- nated	CoordinatedMotion_AC500_V23.lib, MC_CoBlocks_AC500_V23.lib, CMC_Ext_AC500_V23.lib, CMC_Transformationen_AC500_V23.lib	Not yet available	Scheduled for the future
based	ACS350_AC500_V11.lib, ACS350_MC_support_AC500_V11.lib, ACS800_AC500_V11.lib, ACS800_MC_support_AC500_V11.lib, ACSM1_AC500_V11.lib, ACSM1_MC_support_AC500_V11.lib	Not yet available	Scheduled for the future
Temperature control	ADCTRL_AC500_V24.lib, TECT_EXT_AC500_V24.lib, TECT_TEMP_CONTROL_AC500_V24.lib	Not yet available	Scheduled for the future
Pumping (V2 Wa- ter)	PMP_AC500_V25.lib	Pump	
	LogData_AC500_V23.lib	DataLogger	
Logging (V2 Wa- ter)	n.a.	DataLoggerEco	
	n.a.	DataLoggerMulti	
Solar	Solar_AC500_V22.lib, SolarNREL_AC500_V22.lib	Solar	

Drives	ACSDrivesBase_AC500_V20, ACSDrivesComModRTU_AC500_V20.lib, ACSDrivesComModTCP_AC500_V22.lib, ACSDrivesCom- ModTCP_Ext_AC500_V24.lib, ACSDrivesComPB_AC500_V24.lib, ACSDrivesComPN_AC500_V24.lib, DCSDrives_AC500_V24.lib	Drives	
High Availability CS31	HA_CS31_AC500_V23.lib	n.a.	
High Availability Modbus TCP	HAModbus_AC500_V26.lib	HaModbus	
High Availability Modbus TCP	CI52x_AC500_V26.lib	CI52x	
Condition Moni- toring System	CMS_IO_AC500_V24.lib, WAV_FILE_AC500_V24.lib, SP_AC500_V24_App.lib	Not vet available	Scheduled for the future
HVAC	CTRL_AC500_App_V22.lib, HVAC_AC500_App_V22.lib	Not vet available	Scheduled for the future
Process Control Object library	Pco_AC500_V28.lib	Prototype (ask support)	

3.3.5 CAA libraries

V2	V3	Remark
CAA_Async_Man.lib	CAA Async Manager	
CAA_Tick.lib	CAA Tick	
CAA_TickUtil.lib	CAA TickUtil	
CAA_Types.lib	CAA Types	
CAA_Callback.lib	CAA Callback	
CAA_File.lib	CAA File	Most used functions are con- verted to V3. Some functions to work with Archives and on the Disk are not available in V3.

3.3.6 System libraries

Before using function blocks of the CODESYS system libraries, the documentation has to be checked for any changes in functionality. E.g. some rarely used function blocks will be block execution of the PLC application until they are done, which might lead to watchdog exceptions.

V2	V3	Remark
SysLibSockets.lib	SysSocket2	All FUN and FBs are converted to V3
		The V3 library provide much more functionalities then the V2 library
SysLibCom.lib	Com	Same functions are available in V3 library but with different names and interface configu- ration
		No 1:1 use possible
SysLibCallback.lib	n.a.	
SysLibInitLibrary.lib	n.a.	
SysLibMem.lib	SysMem	All FUNs are converted to V3
		The V3 library provide much more functionalities then the V2 library for example Sys- MemIsValidPointer
SysTaskInfo.lib	CmplecTask	Need to be tested
SysLibTime.lib	AC500_Pm SysEvent	Both libraries provide time functions.
SysLibAlarmTrend.lib		Handled by visu library
SysLibEvent.lib	CmplecTask	Same functions are available in V3 library but must be tested
SysLibFile.lib	SysFile	All FUNs are converted to V3
		The V3 library provide much more functionalities then the V2 library
SysLiblecTasks.lib	CmplecTask	Need to be tested!
SysLibPLCConfig.lib	n.a.	
SysLibPlcCtrl.lib	n.a.	See CmpApp
SysLibProjectInfo.lib	СтрАрр	Need to be tested!
SysLibRtc.lib	SysTimeRtc	Need to be tested!
SysLibSem.lib	SysSem_Implementation	Need to be tested!
SysLibStr.lib	StringUtils	

SysLibTasks.lib	SysTask	Need to be tested!
SysLibVisu.lib		Different libraries for V3
SysLib OnlineAccess.lib		Handled by runtime

3.3.7 CODESYS libraries

V2	V3	Remark
standard.lib	Standard	All FUN and FBs are converted to V3 except SEMA (Fb) is not available in V3 Standard library
lecsfc.lib	lecSfc	All FUN and FBs are converted to V3
util.lib	Util	All FUN and FBs are converted to V3 The V3 library provide much more functionalities then the V2 library

3.3.8 Application libraries

V2	V3	Remark
MSSQL_AC500_V24.lib	MsSQL	All FUN and FBs are converted to V3.
		The library is not working with AC500 V3 eCo PLC
MYSQL_AC500_App_V21.lib	MySQL	All FUN and FBs are converted to V3
HTTP_AC500_V24.lib	Http	All FUN and FBs are converted to V3. In addition TLS is sup- ported in the V3 library and further function blocks are available

3.3.9 C-code libraries

Not yet available in AC500 V3.

4 **Conversion of custom libraries**

4.1 Upgrade to V3.5

Before converting the project from V2 to V3 custom libraries needs to be upgraded.

- 1. Open Automation Builder
- 2. Click File → Open Project
- 3. Select CoDeSys library files (before V3.0) and open the library from your hard disk

All supported files Project files (*.project) Library files (*.library) Project archive files (*.projectarchive) CoDeSys project files (before V3.0) CoDeSys library files (before V3.0)

<u>li</u>	

In Automation Builder 2.5.X the opening of library files is by default not possible.

To open a Codesys V2.3 library either open Automation Builder in Profile 2.4 or download and install the CODESYS V2.3 Converter from the codesys store:

https://store.codesys.com/en/codesys-v2-3-converter.html This needs to be installed via package manager (Tools \rightarrow Package Manager)

4. There might be the message, that some libraries are not found. Click No to continue CoDeSys 2.3 to 3.0 Converter

?	Cannot find 1 of 1 libraries. Ch	iange libr	ary directory?	
		Yes	No	1

5. In the popup Device Conversation select Ignore the device

Device Conversion	\times
The original project uses the following device: None	
This device must be replaced by a device in the new file format. What do you want to do?	
Use the following device that has already been installed	
Browse You must select a device by clicking the 'Browse' button.	
Ignore the device. All application specific objects will not be available in the new project.	
Remember this mapping for all future occurrences of that device.	
OK Cancel	

 If other libraries are used in the library, you need to select the corresponding V3 Version. Select use the following library that has already been installed and browse to find the right library in the repository

Library Reference Conversion	×
The original project references the following library: standard.lib ()	
This library reference must be replaced by a reference to a library in the new file format.	
What do you want to do?	
O Convert and install the library as well.	
$\textcircled{\sc online \label{eq:scalar} }$ Use the following library that has already been installed:	
Browse Standard, 3.5.17.0 (System)	
\bigcirc Ignore the library. The reference will not appear in the converted project.	
\bigtriangledown Remember this mapping for all future occurrences of that library reference.	
OK Cancel	

7. Automation Builder will update the library to version 3 and open afterwards. To see the elements, change from Devices to POUs

File	Edit	View	Project	Build	Online	Debug	Tools
1	F 🗄	6	io al	🐴 🕼 [11 列	劉省日	
•							
Devices	;					•	ą 🗙
	Util						-
😤 Dev	ices 🗋	POUs					
						Project use	r: (nobod

- 8. Double click Project information
 - a) Insert your Company, a Title and the Version
 - b) Set the category of the library. The generic Library Category Base description file can

N++

be used or adapted to the needs. Project Information X Automation Builder 2.5 - Basic

Project Information	×	Automation Builder 2.5 - Basic
File Summary Prop	erties Statistics Licensing	Please select one or more of the following categories.
Company Title Version Library Categories	ABB Util_Upgrade 1.0.0.0 Released Application Common	
Library Categories		Application Application
If this project is used as	a library, it will appear under the following categories:	Application Safety Application Safety
	ОК	Cancel

c) Set a namespace for accessing the library



The namespace defines how the internal function blocks and functions are accessed. E.g. when using the namespace TestUtil the function block BLINK can be accessed via the namespace TestUtil.BLINK

d) Set a Placeholder for the library



The placeholder is the name of the library inside the repository. Different versions of a library can then be handled and selected via the placeholder of the library. In many cases the Title of the library is also used as Placeholder.

- e) Fill in Author (optional) and Description (recommended)
- f) By clicking Automatically generate 'Library Information' POUs functions are created which can be used to access the library information via IEC code

Project Information	×		
File Summary Prop	perties Statistics Licensing		
Company	ABB		
Title	Util_Upgrade		
Version	1.0.0.0		
Library Categories	Application Common		
Default namespace	TestUtil		
Placeholder	Util_Upgrade		
Author			
Description	This is a test to update the existing V2.3 util library to V3.5		
The fields in bold letters are used to identify a library.			
Automatically generate 'Library Information' POUs			
Automatically generate 'Project Information' POUs			
	OK Cancel		

- 9. Switch to the tab Properties of the Project Information.
 - a) Click Version string and Remove this Key
 - b) Type Key DocFormat, Type Text, Value reStructuredText and click Add
 - c) Optional the key *LanguageModelAttribute*, Type *Text*, Value *qualified-access-only* can be added



The key LanguageModelAttribute defines if the internal objects can be accessed without the namespace or not.

If the attribute is not set all objects like function blocks, functions, structures, Variables, ... can be accessed without the namespace (normal behavior in V2)

If the attribute is added like described in c) the access is only possible via the namespace e.g. TestUtil.BLINK instead of BLINK

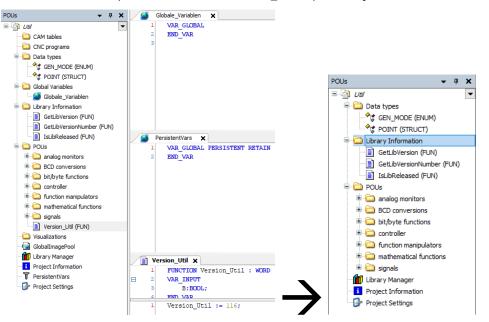
If no qualified access is used all objects inside the library should have a unique prefix. According to the example of the Util library upgrade. If this upgraded library as well as the original library are both added to the project and BLINK is instantiated there will be no error message like in V2.3, but the function block of the library which is more on top of the library manager is used.

d) Click OK to save the changes

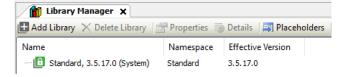
4.2 Changes in POUs, Libraries, ...

After upgrading the library there will be empty or not required objects in the POUs tree which were added during the upgrade can be deleted. Furthermore the existing code should be checked

- 1. Delete all empty folders (Here: CAM tables, CNC programs, Visualization)
- 2. Delete all empty objects (Here Globale_Variablen, GlobalImagePool, PersistentVars)
- 3. Delete all not required POUs (Here Version_Util replaced by GetLibVersion)



4. Open the Library Manager and check that all Library Placeholders are resolved



5. Click Build → Check all Pool Object

The check should run error free and only showing warnings

If the custom library is using non basic CODESYS functionalities additional code changes might be required. The needed steps for different protocols, visualizations, ... are described in the next chapters focusing on the project upgrade.

Solve as many warnings from the build message as possible e.g. by replacing implicit type conversations to explicit type conversations.

6. Check the documentation of the function blocks and change them if required

4.3 Install and share the custom V3 library

In V2 custom libraries could be read and write protected with a password. In V3 there is the differentiation between source and compiled libraries.

A source library *.library is readable by anybody. A source library can be installed into the library repository. If this is the case debugging like step into or breakpoints can be used inside a library

A compiled library *.compiled-library is only binary code. This is neither readable nor editable. There is no way to come from the compiled library back to the source library.

Only compiled libraries should be shared with customers. The access to the source library should be limited to a small group of persons. Please always save the source library somewhere as there is no way to get the source code from a compiled library.

- 1. Save the library
 - a) Click File \rightarrow Save Project as... to save the source library on the hard drive
 - b) Click File → Save Project as Compiled Library... to save the compiled library on the hard drive
- 2. Install the library
 - a) Click Tools \rightarrow Library Repository
 - b) Click Install... and search navigate to the file on the hard drive
 In the drop down menu it might be useful to select All files, as compiled libraries can have different file extensions depending on version, signing, ... of the library

Compiled library files (*.compil \checkmark Compiled library files (*.compiled-library) Compiled library files (*.compiled-library-v3) Library files (*.library) CoDeSys library files (before V3.0) (*.lib)

All files (*.*)

c) Click open to install the library to the library repository

ation	System (C:\ProgramData\AutomationBuilder\AB_Managed Libraries_2.5)	~	Edit Locations
istalled Lil	braries		Install
Company	(All companies)	\sim	Uninstall
A	BB - AC500 pplication Net Base Services 3S - Smart Software Solutions GmbH Common Common System Standard64 System	^	Export
	- Util System - Util_Upgrade ABB - 1.0.0.0	_	Find
<	ii Fieldbus	>	Trust Certificate
Group	by category		Dependencies

Now the placeholder Util_Upgrade can be used on this PC in all projects.



Note: Read more about *Libraries* for AC500 V3 in the Automation Builder online Help. For this, follow the content path:

PLC Automation > PLC Automation with V3 CPUs > Programming with CODESYS> Libraries

After upgrading the custom libraries used in a V2 project the V2 project itself can be upgraded to V3 as described in the next chapter.

5 Conversion of the project

This chapter describes how to upgrade an AC500 V2 project to an AC500 V3 project. The subchapters are focusing on additional adaptions for the hardware configuration, the IEC Application and the visualization.

This chapter is describing the general steps, which need to be done for all projects to upgrade.

The chapters 6 and 7 are then focusing more specific on the fieldbus and SCADA communication protocols and which changes needs to be done there.

- 1. Open the existing AC500 V2 project
- 2. Double click the PLC to open and select Hardware in the menu

Devices 👻 👎 🗙	PLC_AC500_V2 X		
Cockpdate PLC_AC500_V2 (PM564-ETH)	PM564-ETH Parameters	PLC AC500 V2	
Application	PM564-ETH Hardware	PM5xx Type:	PM564-ETH \vee
- 🕼 App - 🕼 OBIO (6DI+6DO+2AI+1AO) = 📶 IO_Bus	I/O mapping list		Change PM / TB type
DC523 (DC523)	Information		
GMILModbus (COM1 - Modbus) GC COM1_None (COM2 - None) COM2_None (COM2 - None) COM2_None (COM2 - None) COM2_None (COM2 - None) GMILMONE GMIL		Change to AC500 V3 PLC PLC AC500 V3 PM5xxx Type:	PM5012-R-ETH 3.5.0.0 PM5012-R-ETH 3.5.0.0 PM5012-T-ETH 3.5.0.0 PM5012-T-ETH 3.5.0.0 PM5032-R-ETH 3.5.0.0 PM5032-T-ETH 3.5.0.0 PM5052-R-ETH 3.5.0.0 PM5052-T-ETH 3.5.0.0 PM5052-T-ETH 3.5.0.0 PM5052-T-ETH 3.5.0.0 PM5052-T-ETH 3.5.0.1 PM5630-ZETH 3.5.0.1 PM5630-ZETH 3.5.0.1 PM5630-ZETH 3.5.0.1 PM5630-ZETH 3.5.0.1 PM5670-ZETH 3.5.0.1 PM5675-ZETH 3.5.0.1

- 3. Click Change to V3 PLC and select the used V3 PLC from the drop-down menu
- 4. Click Create V3 PLC and confirm the dialog

Automat	ion Builder 2.5 - Premium	×
?	The configuration of the selected PLC will be upgraded to V3. Attention: not all configured devices and protocols might be supported by V3 PLC. Unsupported features will be reported during compilation and have to be fixed manually. Do you want to continue?	
	Yes No	

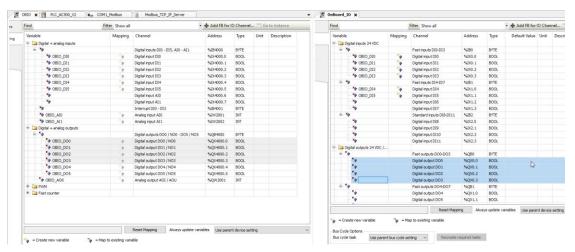
5. The V3 PLC is created in the same project and all devices (including configuration), POUs, visualizations and other elements are automatically added.

5.1 Typical adaptions in the Hardware configuration

5.1.1 IOs

S500 and S500-eCo modules added to the IO bus are also available in the AC500 V3 PLC. All settings and mapping variables remain the same.

Onboard IO settings and mapping, used for AC500-eCo V2 PLCs are not copied and need to be adapted manually. The input and outputs can be just copy & pasted from the AC500 V2 onboard IO to the AC500 V3 onboard IO.



If a AC500-eCo V2 PLC with analog inputs & outputs was used, additional hardware devices are required for these inputs / outputs. These can be realized with spare analog channels in S500(-eCo) modules on the IO bus or additional analog option boards¹ TA5120/ TA5126.

5.1.2 Alarms, Traces & Recipes

After upgrading the project the AlarmConfiguration, Recipe Manager and Sampling_Trace are automatically added to the Project. In case these elements were configured in the AC500 V2 project the elements should be rechecked in the AC500 V3 project.

In case these elements were not used in the existing AC500 V2 program, they can be deleted.

5.1.3 Library Manager

General information regarding library management in AC500 V3 PLCs can be found in chapter 2.3.

Inside the library manager there are already some automatically added libraries depending on the hardware configuration in the project.

Some AC500 V2 libraries might be directly upgraded to a AC500 V3 library with the same functionality. Some libraries have a successor library with similar functionality. For more details check chapter 3.3.

All libraries which could not be resolved needs to be deleted or changed.

¹ Planned for Q4 2022

In almost all cases there are unresolved placeholders in the library manager, which results into error messages. For some of these unresolved placeholders there are already correctly resolved placeholders added, e.g. *DiagloBus* provides the functionality of *BusDiag*.

Library Manager X		
🗄 Add Library 🗙 Delete Library 🚰 Properties 📋 Details 😽 Try to Reload Library 🔻 Downlo	oad Missing Libraries 🛛 🛒	Placeholders 🏼 🎁 Lib
Name Kr	Namespace	Effective Version
AC500_DiagIoBus = Dia Delete Library ABB)	AC500_DiagIoBus	1.2.1.2
AC500_DiagTypes = DiagTypes, 1.2.5.5 (ABB)	AC500_DiagTypes	1.2.5.5
AC500_Ethernet = Ethernet, 1.3.0.7 (ABB)	AC500_Ethernet	1.3.0.7
AC500_Io = Io, 1.2.2.5 (ABB)	AC500_Io	1.2.2.5
AC500_IoDrvEthernet = IoDrvEthernet, 1.0.1.4 (ABB)	AC500_IoDrvEthernet	1.0.1.4
AC500_Pm = Pm, 1.2.6.9 (ABB)	AC500_Pm	1.2.6.9
🖳 🔒 AlarmManager = AlarmManager, 3.5.17.0 (Intern)	AlarmManager	3.5.17.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.17.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.17.0
BusDiag = (not resolved)		

Other libraries need to be exchanged to a V3 successor.

For example for the SysLibMem

SysLibMem = (not resolved)

The easiest way to remove all errors concerning the libraries is to remove all unresolved placeholders from the library manager. After that, Automation Builder will indicate function blocks that cannot be resolved. The names of these function blocks can be entered in the "Add Library" dialog to add the related library. E.g. in case of start typing *SysMemCopy* it will be quickly indicated that the *SysMem* placeholder has to be added.

SysMem	
Match	Library
- 📄 StringBuilder <mark>SysMem</mark> Extending	Datasources
📄 IEC61850_ <mark>SysMem</mark> Cpy	IEC61850Server
📄 <mark>SysMem</mark> Alloc	MathFunctions
SysMemFree	MathFunctions
📄 <mark>SysMem</mark> Swap	SysMem
📄 <mark>SysMem</mark> Set	SysMem
📄 SysMemReallocData	SysMem
📄 <mark>SysMem</mark> Move	SysMem
📄 <mark>SysMem</mark> IsValidPointer	SysMem
	SysMem
🗠 📄 <mark>SysMem</mark> FreeData	SysMem
SysMemForceSwap	SysMem
📄 <mark>SysMem</mark> Cpy	SysMem
📄 <mark>SysMem</mark> Cmp	SysMem
SysMemAllocData	SysMem
VisuFctDatasourcesResourceEntryAllocatorGet_SysMem	VisuElemBase
🗠 🧾 VisuFbDatasourcesResourceEntries_ <mark>SysMem</mark>	VisuElemBase
ReadRecordContent LogObjectsBase	BACnetDefaultImpl
MemoryManager	MemoryUtils
	Pm
APP_MEMORY_SEGMENT	CmpApp

5.1.4 SNTP

AC500 V3 PLC is supporting NTP (Network Time Protocol) instead of SNTP(Simple Network Time Protocol). The difference between these both names are the high level of accuracy for NTP.

Client:

When changing the PLC from AC500 V2 to AC500 V3 the protocol for SNTP **client** is automatically added on the Ethernet interface under "*Protocols* (*Client Protocols*)"

ABB SNTP.project* - Automation Builder 2.5 - Premium		
File Edit View Project Online Debug Tools BACnet Window	Help	
🎦 😅 🔛 😂 🗠 🗠 🛤 🕼 📕 🦄 🧌 🕷 🎁 App [PLC_AC500_V2: J	Application] 🔹 🖾 🐝 🕞 🔳	
Devices – 👎 🛪	SNTP_Client X	
Image: Switzer PCC_ACS00_V2 (PMS83-ETH - TBS21-ETH) Image: PCC_ACS00_V2 (PMS83-ETH - TBS21-ETH) Image: PCC_ACS00_V2 (PMS83-ETH - TBS22-ZETH) Image: PCC_ACS00_V2	Server 1 P address Minimum polling rate (= 64 s) Maximum polling rate (= 1024 s) Server 2 Enable D address Minimum polling rate (= 64 s) Maximum polling rate (= 1024 s)	3 . 0 . 0 6 . . . 0 10 . . . 0 6 . . . 0 10 10
the start the start	Time jumps Time jump threshold (s)	1.0
	Max change	
	 Maximum change offset (s) Start after (clock updates) Ignore after (clock updates) 	3600 3 -1

Note: The settings like IP address, Backup Server etc. are not converted. Please set the NTP settings in the AC500 V3 PLC again.

The settings for the client are similar but in a better readable structure

AC500 V2: SNTP

hi)

AC500 V3: NTP

SNTP_Client 🗙		MTP_Client X	
		Server 1	
		IP address	3.0.0.0
		Minimum polling rate (= 64 s)	6
SNTP Client		Maximum poling rate (= 1024 s)	10
Server 0.0	. 0 . 0	Server 2	
		Enable	
Backup server 0 . 0	. 0 . 0	IP address	0.0.0
Interval	60 s	Minimum polling rate (= 64 s)	6
1 dei vai		Maximum poling rate (= 1024 s)	10
Wait for sync before RUN		Time jumps	
Time to welt for sume	10 s	Time jump threshold (s)	1.0
Time to wait for sync	10 s		
Allow timejumps		Max change	
Alow uncjumps	\leq	Maximum change offset (s)	3600
Allowed threshold for timejumps	60 s	Start after (dock updates)	3
Allowed an earloid for unejumps	3	Ignore after (dock updates)	-1

In both configurations you can set an IP address for the server who will be contacted to receive the current UTC time. Also a Backup Server or Server 2 is supported when Server or Server 1 is not reachable.

Further information can be found in the online help: PLC Automation with V3 CPUs > PLC integration (hardware) > Configuration in Automation Builder for AC500 V3 products > Protocols and special servers > NTP/SNTP protocol > Configuration of the (S)NTP protocol > (S)NTP client configuration

Server:

(tin)

When changing the PLC from AC500 V2 to AC500 V3 the protocol for SNTP **server** need to be added on the Ethernet interface (*ETH1 or ETH2 or both interfaces*).

Note: This protocol is <u>not</u> automatically added when converting the project from AC500 V2 to AC500 V3. The reason is, that the converter does not know at what ETH interface the server needs to be added.

SNTP.project* - Automation Builder 2.5 - Premium		
File Edit View Project Online Debug Tools BACnet Window	Help	
🛅 🚅 🔚 🚭 🗠 🖂 🏘 🎼 川 🦄 🖄 🌆 🖄 🖓 App [PLC_AC500_V2: A	Application] 🝷 😂 🧐 🖒 🔳	
Devices – 4 X	INTP_Server X	
Image: Sympetime Image: Sympetime Image: Sympetime	Access Control	□ 0 · 0 · 0 · 0 24 ☑

The settings for the server are very similar from AC500 V2 to AC500 V3.

AC500 V2: SNTP

SNTP_Server	×							
SNTP Server								
Access control	D	1	0	0	•	0	/ 24	

AC500 V3: NTP

0.0.0.0
24

Access control and Network address have the same behavior. You can set an IP address from a device that is allowed to access and request the time. If the IP is set to 0.0.0.0 means all IP addresses within the same Subnet are allowed. The Subnet mask can be set via /24.

The subnet mask 24 means: 255.255.255.0

The subnet mask 16 means: 255.255.0.0

The NTP Server for AC500 V3 has in addition an Enable checkbox to activate the server.

Further information can be found in the online help: PLC Automation with V3 CPUs > PLC integration (hardware) > Configuration in Automation Builder for AC500 V3 products > Protocols and special servers > NTP/SNTP protocol > Configuration of the (S)NTP proto-col > (S)NTP server configuration

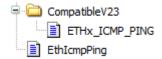
5.2 Typical adaptions in the IEC application

5.2.1 PLCopen conform function blocks

Between AC500 V2 and AC500 V3 the style of the function blocks and functions has changed. The AC500 V3 POUs are according to the PLCopen guidelines.

AC500 V2	AC500 V3
SNAKE_STYLE	CamelCaseStyle
e.g. ETHx_ICMP_PING	e.g. EthlcmpPing
Function blocks have an enable input "EN"	Function blocks are either edge controlled "Execute" input or level controlled "Enable" input
Function block outputs are "DONE", "ERR" and "ERNO"	Function block outputs are "Done", "Error" and "ErrorId"
In case an error occurs DONE and ERR are true	In case an error occurs only ERR is true (ex- clusive)

In some AC500 V3 Libraries there are AC500 V2 compatibility function blocks in a folder called CompatibleV23. For other libraries there is a separate compatibility library with the extension 23 available.



These function blocks can be used to reuse AC500 V2 code in a AC500 V3 PLC without any modification.

The compatibility function blocks are a wrapper around the PLCopen function block with renamed inputs and outputs and the changed behavior in case of errors.

For memory or CPU optimization it is always recommended to use the PLCopen conform AC500 V3 function block instead of the compatibility version.

5.2.2 Modbus addresses %M and %R

As described in chapter 2.1.5 the AC500 V3 PLCs have one single large memory segment instead of several segments like in V2.

The addresses need to be changed accordingly.

%M Addresses

Press CTRL + H for replacing. The goal is to delete the '0.' in the memory addresses.

- Find what: %MX0. Replace with: %MX
- Find what: %MB0. Replace with: %MB
- Find what: %MW0. Replace with: %MW
- Find what: %MD0. Replace with: %MD

Replace		
Find what	%MW0. ~>	Find Next
Replace with	%MW ~>	Find All
Match case	Search up	Replace
Match whole	word Use regular expressions	Replace All
Search	Entire project \checkmark]
Keep modified	Close	

Afterwards all %M memory addresses which were in the 0 segment in V2 have correct V3 addresses.

In case also higher segments were used these need to be changed manually, to avoid overlapping memory areas.

%R Addresses

The %R memory is addressed remanent memory. This area is not existing anymore in AC500 V3.

If the %R area was only used to retain the data and not for Modbus addressing delete the %R address and move the variable to a VAR (GOLBAL) RETAIN PERSISTENT declaration. Either within the Global PersistentVars list or any local declaration.

If the %R area was used for retained Modbus memory, the %R variable must be addressed to any free %M memory address. To retain this %M memory address the pragma {attribute 'noinit'} is used.

VAR_GLOBAL iA AT %MW0 : INT; {attribute 'no_init'} iB AT %MW1 : INT; END_VAR

5.2.3 ACS/DCS Drives Library

There are no AC500 V2 compatible function blocks available. A comparison of AC500 V2 and AC500 V3 functionality and which function blocks should be used can be found in the Online-Help:

PLC Automation with V3 CPUs > Libraries and solutions > ACS/DCS drives libraries > Introduction > Comparison of V2 and V3 drives library

PS553-Drives Library package (V2)		PS5605-Drives Library package (V3)			
Library Name	Function Block	Library	Function Block		
ACSDrivesBase_AC500_V2	ACS3XX_DRIVES_CTRL_BASIC	Not supported – use DrvControlModbusACS	Not supported – use DrvControlModbusACS		
	ACS_DRIVES_CTRL_ENG	ABB_Drives_AC500	DrvControlModbusEng		
	ACS_DRIVES_CTRL_STANDARD		DrvControlModbusACS		
	ACS_DRIVES_CTRL_STANDARD_GEN		DrvControIACS		
	ACS_MOD_READ_N_PRM		DrvModbusRead		
	ACS_MOD_WRITE_N_PRM		DrvModbusWrite		
	ACS_REF_SCALING		DrvScaling		
ACSDrivesComModRTU_AC500_V20	ACS3XX_COM_MOD_RTU	Not supported			
	ACS_COM_MOD_RTU	ABB_Drives_AC500	DrvModbusRtu		
	ACS_COM_MOD_RTU_ENHANCED		DrvModbusRtu		
	ACS_COM_MOD_RTU_GEN	ABB_ModbusRtu_AC500	ModRtuToken		
	ACS_COM_MOD_RTU_GEN_READ_N_PRM		ModRtuRead		
	ACS_COM_MOD_RTU_GEN_WRITE_N_PRM		ModRtuWrite		
			ModRtuReadWrite23		
ACSDrivesComModTCP_AC500_V22	ACS_COM_MOD_TCP	ABB_Drives_AC500	DrvModbusTcp		
	ACS_COM_MOD_TCP_ENHANCED		DrvModbusTcp		
ACSDrivesCom-ModTCP_Ext_AC500_V24	ACS_COM_MOD_TCPx	ABB_Drives_AC500	DrvModbusTcp		
	ACS_COM_MOD_TCPx_ENHANCED		DrvModbusTcp		
DCSDrives_AC500_V24	DCS_DRIVES_CTRL	ABB_Drives_AC500	DrvControlModbusDCS		
	DCS_DRIVES_CTRL_GEN		DrvControIDCS		
ACSDrivesComPN_AC500_V24	ACS_PN_WRITE_N_PRM_DPV1	ABB_Drives_AC500	DrvPnWrite		
	ACS_PN_READ_N_PRM_DPV1		DrvPnRead		
ACSDrivesComPB_AC500_V24	ACS_PB_READ_N_PRM_DPV1	Will be supported in next Release			
	ACS_PB_WRITE_N_PRM_DPV1				
	ACS_COM_PB	Not supported			
	ACS_COM_PB_PZD				
	ACS_PB_READ_PRM_DPV0				
	ACS_PB_WRITE_PRM_DPV0				
		ABB_Drives_AC500	DrvModbusReadWrite23		
		ABB_Drives_AC500	DrvModbusRtuBroadcast		
		ABB_Drives_AC500	DrvControlCANCiA402		

5.3 Typical migration for 3rd party devices

There is a shared device repository for AC500 V2 and AC500 V3. Therefore, all devices that have been available for AC500 V2 can be used with AC500 V3 without any constraints.

6 Reestablishing communication to field devices

6.1 Modbus TCP

The main difference regarding the use of Modbus in AC500 V3 is the difference in the memory structure and therefore in the absolute addressing of the Modbus variables already described in chapter 2.1.5 "Addressable " and chapter 2.1.6 "Modbus addresses / Byte order". Tipps for the conversion of the Modbus addresses are in chapter 5.2.2 "Modbus addresses %M and %R".

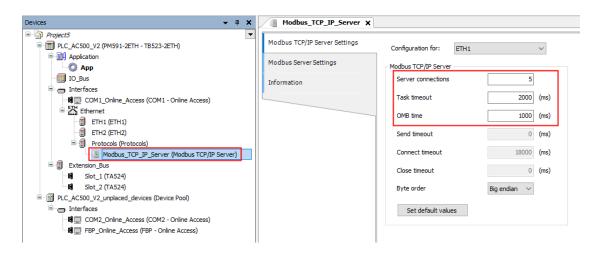
Following other differences have to be taken into account while changing from AC500 V2 to Ac500 V3 with a Modbus TCP connection.

6.1.1 AC500 V2

6.1.1.1 Configuration

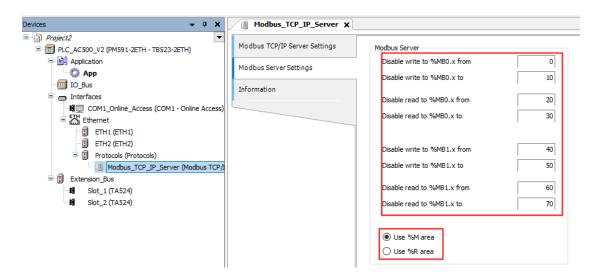
In V2 it is necessary to add a "Modbus_TCP_IP_Server" for both: server and client functionality.

The number of reserved server sockets, task timeout and OpenModBus time can be set in the "Modbus TCP/IP Sever Settings" tab



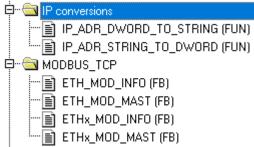
It is possible to protect two memory areas from access (read / write) of other clients access, but it is not possible to prevent the access of the PLC itself.

As the V2 PLCs have a special %R (retain) memory this can be chosen to be used instead of the %M memory area for the Modbus access.



6.1.1.2 Function blocks

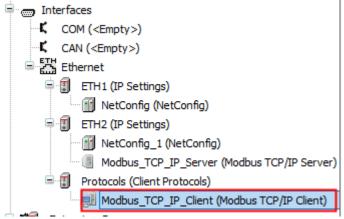
The following main function blocks can be used to create Modbus jobs to read/write from other servers:



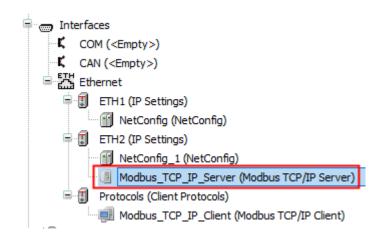
6.1.2 AC500 V3

6.1.2.1 Configuration

To configure the PLC as a Modbus client, the "Modbus_TCP_IP_Client" must be added below the "Protocols" object.



To configure the PLC as a Modbus server, the "Modbus_TCP_IP_Server" must be added below the appropriate ethernet port *ETH1* or *ETH2* (default after conversion is *ETH2*!).



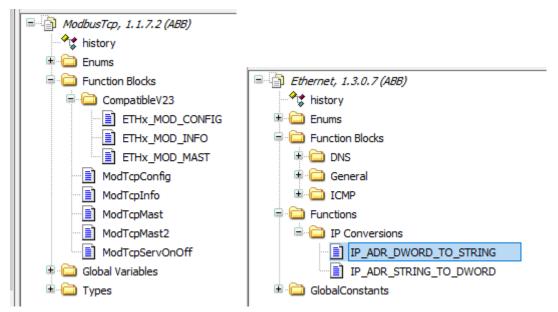
Two memory areas (read / write) can be protected from access from other clients.

Modbus TCP/IP Server Parameters	Parameter	Туре	Value
	👘 🕸 Byte order	Enumeration of BYTE	Big endian
Information	🖤 🖗 Port	WORD(165535)	502
	🔷 🖗 Startup behaviour	Enumeration of BYTE	Active
	Behaviour in state inactive	Enumeration of BYTE	No activity
	🗝 🖗 Disable write to %MBx from	WORD(065535)	0
	 Ø Disable write to %MBx to 	WORD(065535)	10
	Disable read to %MBx from	WORD(065535)	20
	 Disable read to %MBx to 	WORD(065535)	30

Other configurations are made via function blocks ModTcpConfig, ModTcpMast2 and ModTcpServOnOff. See chapter 6.1.2.2 below for details.

6.1.2.2 Function blocks

The following main function blocks can be used to create Modbus jobs to read/write from other servers:



The function blocks beneath the folder "Compatiblev23" can be used directly in the same way as the have been used in AC500 V2.

The ETH_MOD_xx are not available. They must be replaced by the ETHx_MOD_xx, where the only difference is the input "ETH" instead of "Slot".

unction block	K	Remark / part of documentation						
		Confi	Configuration of Modbus TCP for whole PLC					
			RespTimeout	WORD		oonse timeout in ms. Maximum time waiting for a response og sent a request		
ModTcpCo Execute RespTimeout KeepAlive ByteOrder	Done - Busy - Error -	Input	KeepAlive	DWORD	sending a until t	the time is expired. If there is another request within this the timer is restarted. On expiration the connection is		
Port	ErrorID -		ByteOrder	CAA.ENDIANESS	Endianess Endian	of data, little or Big Endian; Modbus Standard is Big		
			Port	WORD	TCP Port			
ModTcpM Execute Eth	Done		RespTimeout	WORD	Slav	e response timeout in ms. Maximum time waiting for a		
-			r and eva	aluate the	e respo			
IPAdr UnitID Fct Addr Nb	Error - ErrorID -		KeepAlive	DWORD		n Modbus Time in ms. A connection to a Server stays n after sending a request until the time is expired. If there is another request with this time, the timer is restarted. On expiration the connection is closed.		
Data RespTimeout		Input	ByteOrder	CAA.ENDIA		aness of data, little or Big Endian; Modbus Standard is Endian		
KeepAlive ByteOrder			Port	WORD	TCP	Port		
Port ResetOnClose			ResetOnClose	e BOOL		et connection when closing it (TRUE) or gaceful close SE) like ModTcpMast		
ConnectTimeout	:		ConnectTime	DWORD	Conr	nect timeout [ms]. Value 0 corresponds to OS default		
						_		
ModTcpSer Execute		Funct	ion bloc	k serves	to set T	CP server to active/passive.		
Execute	Done		Eth	BYTE		Index number of the assigned interface		
Eth	Busy	Input						

Additional functionality and configuration is given by the following function blocks.

6.1.3 Summary of steps for migration from AC500 V2 to AC500 V3

- Add the "MODBUS_TCP_CLIENT" below the "Protocols" object, if Modbus client is needed.
- Check the ethernet port for the server (default is ETH2, should be probably ETH1) if Modbus server is needed.
- In the IEC-Program change the ETH_MOD_yy function blocks to ETHx_MOD_yy function blocks, which are available in same style as in AC500 V2.
- To configure the Modbus behavior of PLC use function block "ModTcpConfig" See chapter 6.1.2.2

- To configure the behavior of dedicated server connections, use the function block "ModTcpMast2", which has special inputs to configure the connection. See chapter 6.1.2.2
- To enable or disable the server functionality of a port use the function block "ModTcpServOnOff". See chapter 6.1.2.2

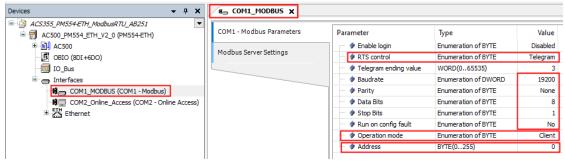
6.2 Modbus RTU

The main difference regarding the use of Modbus in AC500 V3 is the difference in the memory structure and therefore in the absolute addressing of the Modbus variables already described in chapter 2.1.5 "Addressable " and chapter 2.1.6 "Modbus addresses / Byte order". Tipps for the conversion of the Modbus addresses are in chapter 5.2.2 "Modbus addresses %M and %R".

6.2.1 Modbus RTU HW configuration

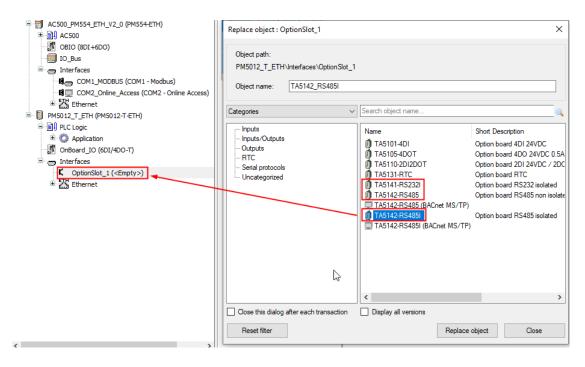
To change from an AC500 V2 PLC with Modbus RTU to an AC500 V3 PLC, the following steps must be performed after the change of the CPU type from AC500 V2 to AC500 V3.

1. Note the communication parameters of the AC500 V2 PLC



RTS control = Telegram \rightarrow RS485 is used. Otherwise RS232. Client mode has always the address "0"

2. For AC500-**eCo V3** insert an **option board** for RS232 or RS485 (with or without isolation) on the dedicated Option Slot.



For AC500 V3 insert COM1 for the COM interface

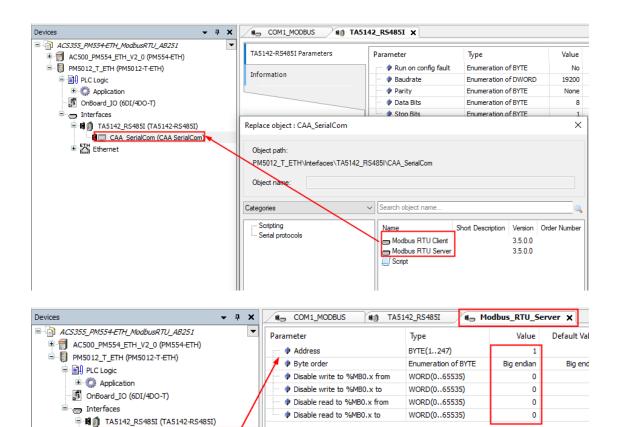
3. Adapte the communication parameters of the option board / COM1 interface

Devices 👻 🕂 🗙	COM1_MODBUS	42_R54851 ×		
ACS355_PM554_ETH_ModbusRTU_AB251 ACS00_PM554_ETH_V2_0 (PM554-ETH)	TA5142-RS485I Parameters	Parameter	Туре	Value [
🖮 🗐 PM5012_T_ETH (PM5012-T-ETH)	Tofo motion	🖗 🖗 Run on config fault	Enumeration of BYTE	No
📮 🗐 PLC Logic	Information	Baudrate	Enumeration of DWORD	19200
Application		🖤 🖗 Parity	Enumeration of BYTE	None
OnBoard_IO (6DI/4DO-T)		🕈 🖗 Data Bits	Enumeration of BYTE	8
interfaces		🔍 🖗 Stop Bits	Enumeration of BYTE	1
응 범 (1) TA5142_RS485I (TA5142-RS485I) - 범 및 CAA_SerialCom (CAA SerialCom) 윤 ఔ Ethernet				

For AC500 V3 also select the communication mode (RS232 or RS485)

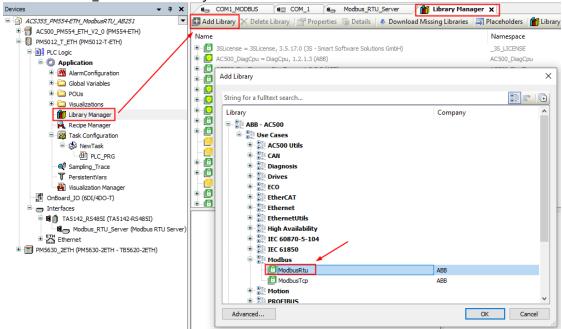
Devices – 4 >	×	/16 0	OM1_MODBUS	∕ n⊒ com_	<u>1 X</u>				
	•	Paramet	ter		Туре			Value	Default Value
		- 🖗 R	un on config fault		Enum	eration of BYTE		No	No
PM5012_1_ETH (PM5012-1-ETH) PM5030_2ETH (PM5630-2ETH - TB5620-2ETH)		- 🖗 Ba	audrate		Enum	eration of DWORD		19200	19200
		🖗 Pa	arity		Enum	eration of BYTE		None	None
IO Bus			ata Bits			eration of BYTE		8	8
Interfaces			top Bits			eration of BYTE		1	1
GOM_1 (COM 1)			low control LC Boot paramete	~	Enum	eration of BYTE	No flow	control	No flow control
CAA_SerialCom (CAA SerialCom)			Serial communic		Enum	eration of BYTE	RS-485	~	RS-232
CAN (<empty>)</empty>							RS-232		
Ethernet							RS-485		
FTH1 (TP Settings)		1						145	

Add a Modbus client or Modbus server.
 If a Modbus server is used the address has to set and the restricted data areas, if needed.



5. Add the AC500_ModbusRtu library

🛤 📖 Modbus_RTU_Server (Modbus RTU Serve



6.2.2 Modbus RTU IEC program adaption

6.2.2.1 Modbus RTU Function blocks

The master function block in AC500 V3 is similar to AC500 V2, but in PLCopen style and has an additional input "Endian" to select the endinaness (byte order)

AC500 V2:

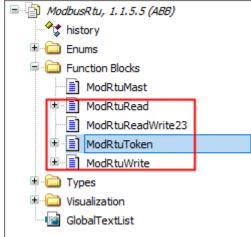
COM_MOD_MAST				
EN : BOOL COM : BYTE SLAVE : BYTE FCT : BYTE TIMEOUT : WORD ADDR : WORD NB : WORD DATA : DWORD	DONE : BOOL ERR : BOOL ERNO : WORD			

AC500 V3:

ModRtuMast	
—[Execute BOOL := FALSE]	BOOL Done
[Com BYTE := 1]	BOOL Busy
— [Endian CAA.ENDIANESS := CAA.ENDIANESS.BIG]	BOOL Error
[Serv BYTE := 0]	ERROR_ID ErrorID
[Fct BYTE := 0]	
[Timeout WORD := 2000]	
[Addr WORD := 0]	
-[Nb WORD := 0]	
[Data DWORD := 0]	

Additional function blocks in AC500 V3:

The AC500 V3 library contains some more valuable function blocks that encapsulate the necessary step chain to cyclically read/write data from one or several servers:



Using the ModRtuToken together with one or more of the blocks ModRtuRead, ModRtuWrite or ModRtuReadWrite23 establishes cyclic read/write jobs of all servers connected to the same COM line.

ModRtuToken	
[Enable BOOL := FALSE]	BOOL Busy
[Com BYTE := 1]	BOOL Error
[Serv BYTE := 1]	ERROR_ID ErrorID
-[TimeOut WORD := 1000]	BOOL ModMastErrorAct
[ReconnectPause WORD := 0]	ERROR_ID ModMastErrorIDLast
LineToken ModRtuTokenType	BOOL WarnAct
— ServData ModRtuGenDevDataType	WARNING_ID WarnIDLast
	BOOL JobDone
	BOOL JobBusy
	BOOL Online

Connection between the different servers is made via the common variable connected to the input "LineToken".

Connection to the read/write blocks within one server is made via the common variable "ServerData".

6.2.2.2 Modbus data in AC500 V3

If the AC500 V3 is used as a Modbus server, the absolute addressing of the variables need to be checked / changed.

If special conversion of received or send data were made in AC500 V2, those have to be checked due to the different byte order in AC500 V3 PLCs.

See chapter 2.1.5 "Addressable " and chapter 2.1.6 "Modbus addresses / Byte order"

6.3 Profinet

6.3.1 General

All devices and couplers are kept as long as a AC500 V3 CPU is selected, that has enough communication modules slots available (no AC500-eCo V3).

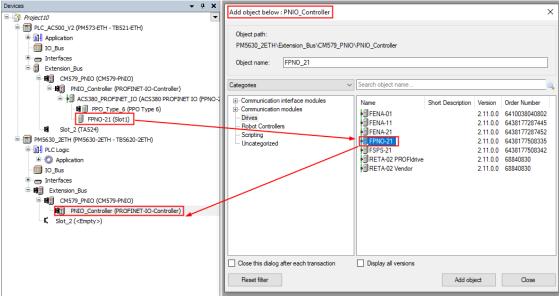
Exceptions are:

- There are no DriveManager editors, as they are not supported by the AC500 V3 PLCs. See chapter 6.3.2 how to keep the data exchange functionality.
- CI504 and CI506 are not yet supported by AC500 V3.

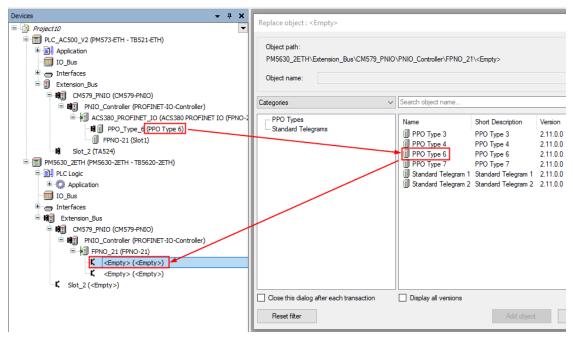
6.3.2 DriveManager

The DriveManager elements for ACS/DCS drives are not supported by AC500 V3. To keep the data exchange functionality, it's necessary to do some manual steps for each drive.

Insert a FPNO-21 or FENA-x1 node below the Profinet_Controller according to what was selected in the AC500 V2 PLC.



Then add the same PPO-Type as used in the AC500 V2 PLC to the FPNO-21 or FENA-x1 node.



Then copy all variable mappings from the AC500 V2 PLC to the AC500 V3 PLC. Best to use the I/O mapping list.

Mark the whole column with all variables, copy them, go to the V3 I/O mapping list, mark the whole column and paste the variables.

ACS380_PROFINET_IO	× V2 PLC				✓ #◎ PPO_Type_6 × V3 PL	C			
Process data	່າ ເ	appings 🛛 🔻			Diagnosis	🤅 🔈 . ୯ 🛛 🗙 Clear ma	appings 🛛 🔻		
	Object Name	Variable	Channel	Address		Object Name	Variable	Channel	Address
Drive Management	PPO_Type_6	ACS380_PROFINET_IO_SW16bit	Status	%IW1.0	General	PPO_Type_6		Status	%IW0
Monitoring	PPO_Type_6	ACS380_PROFINET_IO_Act116bit	Speed Actual	%IW1.1	PNIO Module I/O Mapping	PPO_Type_6		Speed Actual	%IW1
Homeoning	PPO_Type_6		Actual PZD3	%IW1.2	E	PPO Type 6	-	Actual PZD3	%IW2
General	PPO_Type_6		Actual PZD4	%IW1.3	PNIO Module IEC Objects	PPO_Type_6		Actual PZD4	%IW3
	PPO_Type_6		Actual PZD5	%IW1.4	ii	PPO_Type_6		Actual PZD5	%IW4
Options	PPO_Type_6		Actual PZD6	%IW1.5	I I/O mapping list	PPO_Type_6		Actual PZD6	%IW5
	PPO_Type_6		Actual PZD7	%IW1.6		PPO_Type_6		Actual PZD7	%IW6
PROFINET IO Device	PPO_Type_6		Actual PZD8	%IW1.7	Information	PPO_Type_6		Actual PZD8	%IW7
I/O mapping list	PPO_Type_6		Actual PZD9	%IW1.8		PPO_Type_6		Actual PZD9	%IW8
tyo mapping list	PPO_Type_6		Actual PZD10	%IW1.9	0	PPO_Type_6		Actual PZD10	%IW9
Log	PPO_Type_6	ACS380_PROFINET_IO_CW16bit	Command	%QW1.0		PPO_Type_6		Command	%QW0
· ·	PPO_Type_6	ACS380_PROFINET_IO_Ref116bit	Speed Reference	%QW1.1	E .	PPO_Type_6		Speed Reference	%QW1
Information	PPO_Type_6		Reference PZD3	%QW1.2	E.	PPO_Type_6		Reference PZD3	%QW2
	PPO_Type_6		Reference PZD4	%QW1.3	C.	PPO_Type_6		Reference PZD4	%QW3
	PPO_Type_6		Reference PZD5	%QW1.4	0	PPO_Type_6		Reference PZD5	%QW4
	PPO_Type_6		Reference PZD6	%QW1.5	0	PPO_Type_6		Reference PZD6	%QW5
	PPO_Type_6		Reference PZD7	%QW1.6		PPO_Type_6		Reference PZD7	%QW6
	PPO_Type_6		Reference PZD8	%QW1.7		PPO_Type_6		Reference PZD8	%QW7
	PPO_Type_6		Reference PZD9	%QW1.8	E	PPO_Type_6		Reference PZD9	%QW8
	PPO_Type_6		Reference PZD10	%QW1.9		PPO_Type_6		Reference PZD10	%QW9

6.3.3 Bus cycle

The bus cycle can be adapted in AC500 V3 to a specific task. This will be set by the parameter "Bus cycle task" of the Profinet Master IO-Controller

Extension_Bus		
🖹 🕼 CM579_PNIO (C	M579-PNIO)	
😑 🛋 CM579_Mas	ter (PROFINET-IO-Co	ntroller)
🛯 CM579_Master 🗙		
Diagnosis	Bus Cycle Options	
	Bus cycle task	Use parent bus cycle setting \sim
General		Use parent bus cycle setting NewTask
PROFINET-IO-Controller I/O Mapping	L	
PROFINET-IO-Controller IEC Objects		
I/O mapping list		
Information		

6.3.4 Cyclic data exchange

After the conversion from AC500 V2 to AC500 V3 PLC the Profinet controller and device functionality regarding it's cyclic data exchange is kept as in V2 and can be used without any changes, as long as all needed variables are mapped and used within the IEC program code.

Absolute addresses might change and should not be used in the IEC program code.

6.3.5 Acyclic data exchange

The acyclic data exchange needs to be adapted slightly as the function blocks have changed to PLCopen style and have some improvements in outputs Status, AddVal1 and AddVal2.

V2		V3	
PNIO_RE/	D_EXT	PnioCnt	
EN SLOT DEV_NAME DEV_API DEV_SLOT DEV_SUB_SLOT DEV_IDX DATA DATA_MAX	DONE ERR ERNO STATUS ADD_VAL1 ADD_VAL2 DATA_LEN	Execute Slot DevName DevApi DevSlot DevSubSlot DevIdx Data DataMax	Done Busy Error ErrorID AddErrNo Status AddVal1 AddVal2 DataLen
PNIO_WRIT	TE EXT	PnioCntr	lWrite
EN	DONE-	Execute	Done
SLOT	ERR-	Slot	Busy
DEV_NAME	ERNO-	DevName	Error
DEV_API	STATUS-	- DevApi	ErrorID
DEV_SLOT	ADD_VAL1	DevSlot	AddErrNo
DEV_SUB_SLOT	ADD_VAL2	DevSubSlot	Status
DEV_IDX		DevIdx	AddVal1
DATA		Data	AddVal2
DATA_LEN		DataLen	

6.3.6 Bus control

The bus control function blocks have kept the functionality but are now available in PLCopen style.

V2	V3
PNIO_CNTL_START_COM EN DONE SLOT ERR TOUT ERNO ADD_ERNO	PnioCntrlStartCom Execute Done - Slot Busy - Error - ErrorID - AddErrNo -
PNIO_CNTL_STOP_COM EN DONE- SLOT ERR- RESET_IO ERNO- TOUT ADD_ERNO-	PnioCntrlStopCom Execute Done - Slot Busy - Error - ErrorID - AddErrNo -

6.3.7 Diagnosis

The AC500 V3 diagnosis system gives detailed information about the bus state and S500 module messages (see chapter 2.5). Additional bus scans can be done via Automation Builder.

The Profinet diagnosis function block, which are used in AC500 V2 must be adapted to the function blocks in AC500 V3, which are now available in PLCopen style and partly include functionality of multiple blocks from AC500 V2 in one block in AC500 V3.

/2		V	3	
PNIO_S	TATE		PnioCntrlG	etCntrlState
			Execute	Done
EN : BOOL	DONE : BOOL-	-	Slot	Busy
SLOT : BYTE	ERR : BOOL-			Error
MST_STATE	ERNO: WORD- PNIO_MST_STATE_TYPE-			ErrorID
_	PNIO_COMM_ERNO_TYPE			AddErrNo
_	NUM_ERRS : DWORD-		Co	mmunicationCOS
				MstState
	1			CommErno
PNIO_SYS_DIAG EN DONE				NumErrs
SLOT ERR	-			Version
ERNO				Watchdog
DEV_STATE				SlaveState
NUM_CFG			N	umConfigSlaves
NUM_ACT			N	umActiveSlaves
NUM DIAG				NumDiagSlaves

	PnioCntrlGetDevState
PNIO_DEV_INFO_EXT	Execute Done
EN DONE	
	Slot Busy
SLOT ERR	DevName Error
DEV_NAME ERNO	
	ErrorID -
MAC	AddErrNo
IP-	
VENDOR-	Mac -
	Ip-
DEVICE	Vendor -
ALARM-	vendor
	Device -
ACTIVE-	Alarm-
FLAGS-	
	Active -
	Flags -
	PnioCntrlGetDevIM0Data
PNIO_IM0	Execute Done
HEN DONE	Slot Busy
	1
HSLOT ERR	21101
DEV_NAME ERNO	DevSlot ErrorID
	AddErrNo
-DEV_SLOT VENDOR_ID-	
ORDER_ID	VendorId -
SER_NO-	OrderId -
	SerNo
HW_REV	
SW_REV	HwRev —
REV_CNT-	SwRev -
PROFILE_ID	RevCnt -
PROFILE_SPEC	ProfileId -
	ProfileSpec -
IM_VERSION-	
IM_SUPPORTED	ImVersion -
	ImSupported -
PNIO_DEV_ALARM	
EN DONE	
SLOT ERR	
DEV_NAME ERNO	
DEV_SLOT-	
DEV_SUB_SLOT-	
DEV MODULE	
DEV_MODOLE	Please use standard diagnose of node name.
PRIO	-
	<node_name>.diag</node_name>
	Or use generic function block from diasecto
	Or use generic function block from dignosis
DATA_LEN	library
NEW-	
ACTIVE-	
PNIO_DEV_SPECIFIER	
SPECIFIER SEQUENCE	
CHANNEL—	
MANUFACT-	
SUB_MOD-	
RES-	
AR_STATE-	
PNIO_DEV_DIAG	Please use standard diagnose of node name.
EN DONE	
-SLOT ERR-	chada namas dias
	<node_name>.diag</node_name>
DEV_NAME ERNO	
-DATA FLAGS-	Or use generic function block from dignosis
DATA_LEN-	library
	IIDIALA
BAINTEEN	norary
DANGEER	

6.4 Profibus

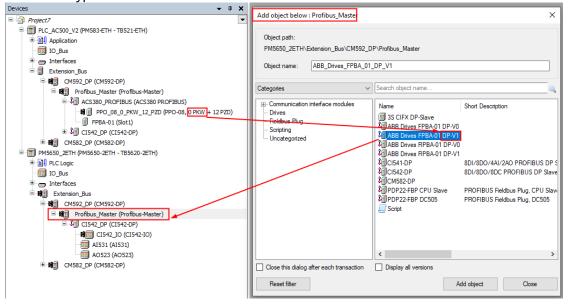
6.4.1 General

All devices and couplers are kept as long as a AC500 V3 CPU is selected, that has enough communication modules slots available (no AC500-eCo V3). The only exception is that there are no DriveManager editors, as they are not supported by the AC500 V3 PLCs. See chapter 6.4.2 how to keep the data exchange functionality.

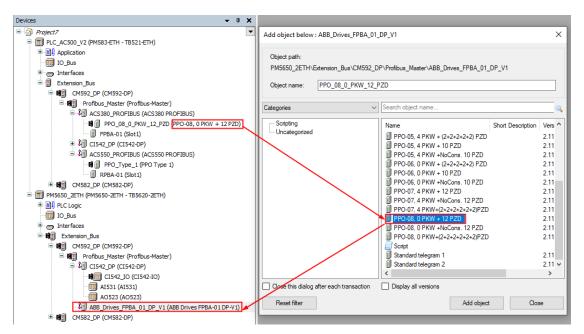
6.4.2 DriveManager

The DriveManager elements for ACS/DCS drives are not supported by AC500 V3. To keep the data exchange functionality it's necessary to do some manual steps for each drive.

Insert a FPBA-01 or RPBA-01 node below the Profibus_Master. If the PPO-Type used in the V2 PLC is with **0 PKW** insert a FPBA-01 or RPBA-01 **DPV1** If the PPO-Type used in the V2 PLC is with **4 PKW** insert a FPBA-01 or RPBA-01 **DPV0**



Then add the same PPO-Type as used in the V2 PLC to the FPBA-01 or RPBA-01 node.



Then copy all variable mappings from the AC500 V2 PLC to the AC500 V3 PLC. Best to use the I/O mapping list.

Mark the whole column with all variables, copy them, go to the AC500 V3 I/O mapping list, mark the whole column and paste the variables.

General	່າ ເ 🗙 Clear map	ings 🔻			Diagnosis	🗄 🤊 ເ 🗙 Clear mappin	js 🛛 🔻		
	Object Name	Variable	Channel	Address 1		Object Name	Variable	Channel	Addre
DP-Module I/O Mapping	PPO_08_0_PKW_12_PZ	ACS380_PROFIBUS_SW16bit	nput0 - Word0	%IW1.0 W	General	PPO_08_0_PKW_12_PZD		Input0 - Word0	%IW1
I/O mapping list	PP0_08_0_PKW_12_PZ	0	nput0 - Word0 - Bit0	%IX1.0.0 B	DP-Module I/O Mapping	PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit0	%IX28
To mapping ist	PP0_08_0_PKW_12_PZ)	nput0 - Word0 - Bit1	%IX1.0.1 B	br Hodale yo Happing	PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit1	%DX2
Information	PP0_08_0_PKW_12_PZ		nput0 - Word0 - Bit2	%DX1.0.2 B	DP-Module IEC Objects	PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit2	%IX2
	PPO_08_0_PKW_12_PZ		nput0 - Word0 - Bit3	%IX1.0.3 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit3	%IX2
	PPO_08_0_PKW_12_PZ	0	nput0 - Word0 - Bit4	%IX1.0.4 B	I/O mapping list	PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit4	%IX2
	PP0_08_0_PKW_12_PZ	0	nput0 - Word0 - Bit5	%IX1.0.5 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit5	%IX2
	PP0_08_0_PKW_12_PZ	0	Input0 - Word0 - Bit6	%IX1.0.6 B	Status	PP0_08_0_PKW_12_PZD		Input0 - Word0 - Bit6	%IX2
	PP0_08_0_PKW_12_PZ	0	Input0 - Word0 - Bit7	%IX1.0.7 B	Information	PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit7	%IX2
	PP0_08_0_PKW_12_PZ)	Input0 - Word0 - Bit8	%IX1.1.0 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit8	%D(2
	PP0_08_0_PKW_12_PZ)	Input0 - Word0 - Bit9	%IX1.1.1 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit9	%DC
	PPO_08_0_PKW_12_PZ		nput0 - Word0 - Bit10	%IX1.1.2 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit10	%IX
	PP0_08_0_PKW_12_PZ		nput0 - Word0 - Bit11	%IX1.1.3 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit11	%IX
	PP0_08_0_PKW_12_PZ		nput0 - Word0 - Bit12	%IX1.1.4 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit12	%IX
	PP0_08_0_PKW_12_PZ	0	Input0 - Word0 - Bit13	%IX1.1.5 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit13	%IX
	PP0_08_0_PKW_12_PZ	0	Input0 - Word0 - Bit14	%IX1.1.6 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit14	%D
	PP0_08_0_PKW_12_PZ)	Input0 - Word0 - Bit15	%IX1.1.7 B		PPO_08_0_PKW_12_PZD		Input0 - Word0 - Bit15	%IX
	PP0_08_0_PKW_12_PZ	ACS380_PROFIBUS_Act116bit	Input0 - Word1	%IW1.1 W		PPO_08_0_PKW_12_PZD		Input0 - Word1	%IV
	PP0_08_0_PKW_12_PZ	0	nput0 - Word1 - Bit0	%IX1.2.0 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit0	%IX
	PP0_08_0_PKW_12_PZ		nput0 - Word1 - Bit1	%IX1.2.1 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit1	%IX
	PP0_08_0_PKW_12_PZ		Input0 - Word1 - Bit2	%IX1.2.2 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit2	%IX
	PPO_08_0_PKW_12_PZ		nput0 - Word1 - Bit3	%IX1.2.3 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit3	%IX
	PP0_08_0_PKW_12_PZ	0	nput0 - Word1 - Bit4	%IX1.2.4 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit4	%IX
	PP0_08_0_PKW_12_PZ		Input0 - Word1 - Bit5	%IX1.2.5 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit5	%IX
	PP0_08_0_PKW_12_PZ	0	nput0 - Word1 - Bit6	%IX1.2.6 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit6	%IX
	PP0_08_0_PKW_12_PZ	0	nput0 - Word1 - Bit7	%IX1.2.7 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit7	%IX
	PP0_08_0_PKW_12_PZ		nput0 - Word1 - Bit8	%IX1.3.0 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit8	%IX
	PPO_08_0_PKW_12_PZ		nput0 - Word1 - Bit9	%IX1.3.1 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit9	%IX
	PP0_08_0_PKW_12_PZ	0	nput0 - Word1 - Bit10	%IX1.3.2 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit10	%IX
	PP0_08_0_PKW_12_PZ		nput0 - Word1 - Bit11	%IX1.3.3 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit11	%IX
	PP0_08_0_PKW_12_PZ		nput0 - Word1 - Bit12	%IX1.3.4 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit12	%IX
	PPO_08_0_PKW_12_PZ		nput0 - Word1 - Bit13	%IX1.3.5 B		PPO_08_0_PKW_12_PZD		Input0 - Word1 - Bit13	%DC

6.4.3 Bus cycle

The bus cycle can be adapted in AC500 V3 to a specific task. This will be set by the parameter "Bus cycle task" of the CM592_DP in the CM592-DP I/O Mapping section.

Extension_Bus		
🕫 CM592_DP 🗙 🌒 Profibus_	Master	
CM592-DP Parameters	-Bus Cycle Options Bus cycle task	
CM592-DP I/O Mapping	bus cycle task	Use parent bus cycle setting Use parent bus cycle setting DefaultTask
CM592-DP IEC Objects		
Status		
Diagnosis		
Information		

6.4.4 Cyclic data exchange

After the conversion from AC500 V2 to AC500 V3 PLC the Profibus master and slave functionality regarding it's cyclic data exchange is kept as in AC500 V2 and can be used without any changes, as long as all needed variables are mapped and used within the IEC program code.

Absolute addresses might change and should not be used in the IEC program code.

6.4.5 Acyclic data exchange

The acyclic data exchange needs to be adapted slightly as the function blocks have changed to PLCopen style.

/2	V3
DPV1_MSAC1_READ EN : BOOL DONE : BOOL SLOT : BYTE ERR : BOOL SLV : BYTE ERNO : WORD SLV_SLOT : BYTE ERNO1 : BYTE SLV_IDX : BYTE ERNO2 : BYTE LEN : BYTE DATA_LEN : BYTE DATA : DWORD	CM592DFV1Masc1Read Execute Done - Device Busy - SlaveAddr Error - SlaveSlot ErrorID - SlaveIndex AddErrorID - DataLen ErrNo1 - Data ErrNo2 - DataLenRead -
DPV1_MSAC1_WRITE =EN : BOOL DONE : BOOL =SLOT : BYTE ERR : BOOL =SLV : BYTE ERNO : WORD =SLV_SLOT : BYTE ERNO1 : BYTE =SLV_IDX : BYTE ERNO2 : BYTE =LEN : BYTE DATA_LEN : BYTE =DATA : DWORD	CM592DPV1Masc1Write Execute Done Device Busy SlaveAddr Error SlaveSlot ErrorID SlaveIndex AddErrorID DataLen ErrNo1 DataLenWrite DataLenWrite
DPM_READ_INPUT -EN : BOOL DONE : BOOL -SLOT : BYTE ERR : BOOL SLV : BYTE ERNO : WORD -DATA : DWORD DATA_LEN : BYTE	CM592ReadInput Execute Done Device Busy SlaveAddr Error DataLen ErrorID Data AddErrorID DataLenRead
DPM_READ_OUTPUT -EN : BOOL DONE : BOOL- -SLOT : BYTE ERR : BOOL- -SLV : BYTE ERNO : WORD- DATA : DWORD DATA_LEN : BYTE-	CM592ReadOutput Execute Done Device Busy SlaveAddr Error DataLen ErrorID Data AddErrorID DataLenRead

6.4.6 Bus control

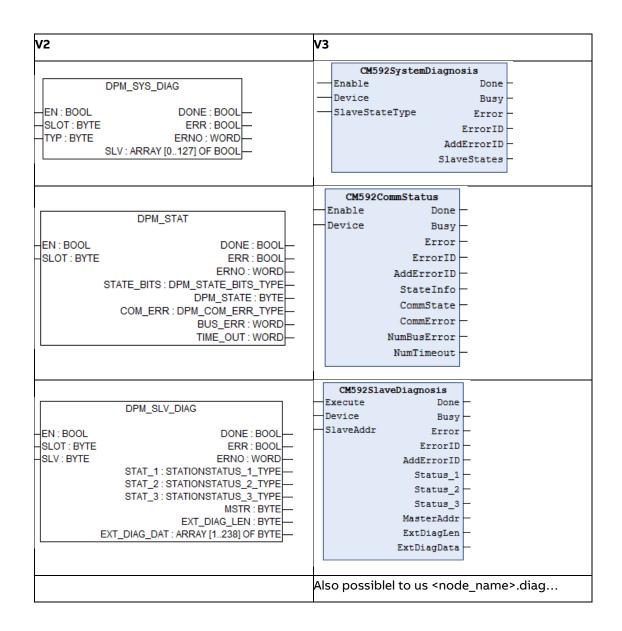
The bus control function blocks have kept the functionality but are now available in PLCopen style.

V2	V3
DPM_CTRL =EN : BOOL DONE : BOOL	CM592Control Execute Done Device Busy SlaveAddr Error GroupSelect ErrorID Unfreeze AddErrorID Freeze Unsync Sync Sync

6.4.7 Diagnosis

The V3 diagnosis system gives detailed information about the bus state and S500 module messages (see chapter 2.5). Additional bus scans can be done via Automation Builder.

The profibis diagnosis function blocks, which are used in AC500 V2 must be adapted to the function blocks in AC500 V3, which are now available in PLCopen style.



6.5 CS31 Bus connections

There is no CS31-Bus available in AC500 V3 nor AC500-eCo V3 CPUs yet.

Please check possible availability in the future.

For the time being the IO-Modules used on a CS31 Bus must be transferred to another decentralized communication interface.

Possible Solutions:

1. Using Profibus CI541-DP or CI542-DP with AC500 V3 (serial cable might be kept)

- 2. Using Modbus TCP CI521-MODTCP or CI522-MODTCP with AC500-eCo V3 or AC500-eCo (can be used with eCo. Needs special engineering)
- 3. Using Profinet CI501-PNIO or CI502-PNIO with AC500 V3

7 Reestablishing communication to SCADA devices

7.1 Symbol File

7.1.1 AC500 V2 PLC



This chapter only describes the steps how to configure the symbol files for AC500 V2 PLCs. The next chapter 7.1.2 describes the configuration for AC500 V3 PLCs.

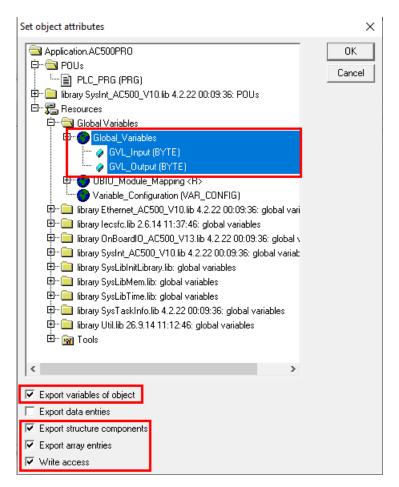
Configure Symbol File

Symbol includes the items(variables) which exchanges with PLC, this is needed for OPC communication with Panel, SCADA, etc.

Open the Symbol configuration (Project->Options->Symbol configuration) to choose 'Dump symbol entries', and click the button 'Configure symbol file...'

Options		×
Options Category: Load & Save User Information Editor Desktop Colors Directories Log Build Passwords Source download Symbol configuration Database-connection Macros	Dump symbol entries Dump XML symbol table Configure symbol file	Cancel
Macros		

Only choose the variables which should be communicated as symbol. Then check the options except for the item 'Export data entries '



Confirm the settings with pushing the OK button. Then we can rebuild the project.

Download Symbol file to PLC

Open the target settings from resources tab and check the 'Download symbol file '.

	1 +	Task configuration			
🔚 Resources	Target Settings				×
🛱 🗠 🔂 Global Variables					
	Configuration: AC50	0 PM566-ETH 2.x		-	
OBIO_Module_Ma					
🦾 🕐 Variable_Configura	Target Platform Memo	ry Layout General Nei	twork functionality Visualizatio	on	
🗄 🗀 library Ethernet_AC500	□ 1/0-Configuration —				
🗄 🗀 library lecsfc.lib 2.6.14	_				
🖶 💼 library OnBoardIO_AC5	🗹 Configurable				
🗄 🗀 library SysInt_AC500_V					
🖶 💼 library SysLibInitLibrary.					
🖽 💼 library SysLibMem.lib: g		dun multipertaine	Download work of Go		0UT
🗄 🕮 library SysLibTime.lib: g			Download symbol file		OUT as reference
🗄 🖆 library SysTaskInfo.lib 4	🔲 Single task in m	ulti-tasking	Symbol config from INI file	🔽 Initialize i	nputs
🗄 🗀 library Util.lib 26.9.14 1	🔲 Byte addressing	mode 🔽	PLC Browser	🔽 Load boo	ot project automatically
🗄 🖓 Tools	🔽 Initialize zero	v	Trace	🔲 SoftMotic	in
🕅 Alarm configuration	🔽 Online Change			🔲 Retain fo	rcing 🗖 Save
🗂 🎢 Library Manager	Update unused	U0.			-
🞒 Log		//05			
💼 PLC - Browser					
🏢 PLC Configuration					0K 0 1
····· 🞑 Sampling Trace			_	Default	OK Cancel
🗝 🚔 Target Settings					
📟 🎆 Task configuration					
	anager				
- 🕂 🛠 Workspace					

Go online and download the project into PLC. After that save and close the Codesys IEC application editor.

We can export the symbol file with text format which can be understood by engineer.

Devices		•	ч х		BIO 🗐 PLC_AC500_V	2
OPCDA_eCo OPCDA_eCo OPCDA_eCo O OPCDA_eCo O	6-ETH)		•		Parameter Type	ratior
Application	Paste Rename					
□ Interfaces	Add object Update objects					
	Add Folder Edit Object					
	Compare Objects Import	•				
	Export SVN	•			C application (folder) C application (library)	
	Manage additional files for PLC Create configuration data		P	'L(C application (HMI project) C Hardware configuration	
			В	0	ot project and firmware (SD card) ot project (folder)	
				-	mbol file mware (SD card)	

Then we can check if the symbol file is generated correctly.

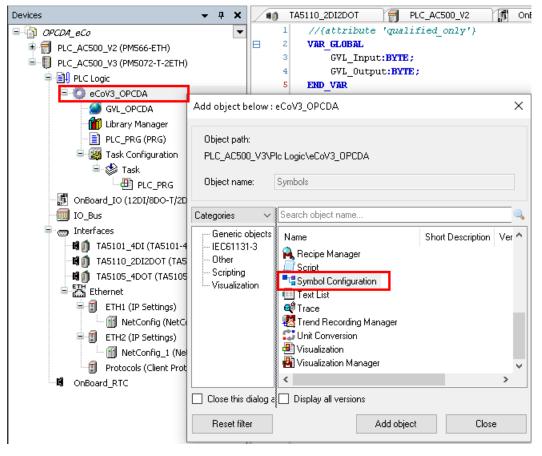
Name	Date modified	Туре	Size				
Application.SYM	02.08.2022 15:16	SYM File	1 K	В			
Cillicore) DEU2	19619\Desktop\symbol\App	lication SVM - Not	anad				
				Diversion W/s			
	n View Encoding Langu			Plugins Wir			
	3 (2 🖨 🖌 🖻 👘 3	9 C m %	🤜 🔜 🖓 🖼	Ep 11 🎼 🛛			
🔚 Application.SYM	×						
1 ;Versio	on=2						
2 ;Projec	ctId=74186						
3 ;Checks	3 ;Checksum=0						
4 ;Filesi	;Filesize=O						
5 .GVL_Ir	.GVL Input:BYTE:5:1958:1:b:16#02000040						
6 .GVL Ou	.GVL_Output:BYTE:5:1959:1:b:16#02000040						
7							



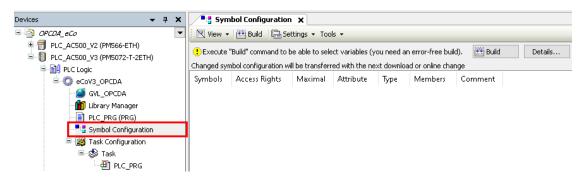
Sometimes the symbol file looks different than really configured, e.g. more symbols than expected. In such a case please repeat the steps again.

7.1.2 AC500 V3 PLC

Add the object Symbol Configuration to the Application object in the Automation Builder device tree.

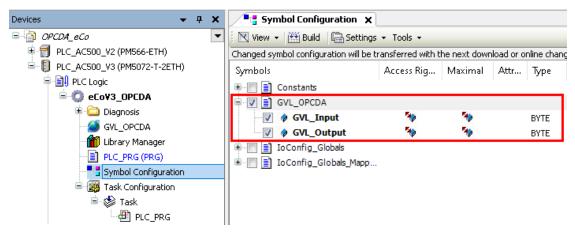


The symbol configuration editor appears. Keep the default settings and click Add button.



In symbol configuration editor, execute 'Build' command to be able to select variables.

Then all the project items(variables) will be displayed in the window. Select only the desired items. These are needed e.g. for communication with a CP600 Panel, OPC DA or OPC UA client.



Go online and download the project into PLC. The symbol file is automatically generated by a build command of an AC500 V3 project when a symbol configuration exists and is stored with the name extension XML next to the project file.

OPCDA eCo.PLC AC500 V3.eCoV3 OPCDA.205667bc-02bf-	4a51-93a4-a00c9d1f 🥝	03.08.2022 13:45	COMPILEINFO File	5.292 KB
COPCDA_eCo.PLC_AC500_V3.eCoV3_OPCDA.xml	\odot	03.08.2022 13:45	Microsoft Edge H	1 KB
ABB OPCDA_eCo.project	Ø	03.08.2022 13:32	Automation Build	736 KB
OPCDA_eCo.project.~u	\odot	03.08.2022 13:58	~U File	1 KB
OPCDA_eCo_project.precompilecache	\odot	03.08.2022 13:32	PRECOMPILECAC	155 KB
OPCDA_eCo-AllUsers.opt	\odot	03.08.2022 13:32	OPT File	1 KB
	0	AD AD 1011 10.00	ODT DU-	100 1/0

The symbol file can be opened with any text editor and checked if the entries are correct.

🔚 OPC	CDA_eCo.PLC_AC500_V3.eCoV3_0PCDA.xml 🔀	
1	xml version="1.0" encoding="utf-8"?	
2	<pre>F<symbolconfiguration xmlns="http://www.3s-software.com/schemas/Symbolconfiguration.xsd"></symbolconfiguration></pre>	
3	📮 🗘 <header></header>	
4	<version>3.5.14.0</version>	
5	<pre><symbolconfigobject 0pcda_eco"="" appname="eCoV3_0PCDA" compiler="3</pre></td><td>.5.17</td></tr><tr><td>6</td><td><pre><ProjectInfo name=" devicename="PLC_AC500_V3" libversion="3.5.17.0" runtimeid="3.5.17.10" version="4.0.0.0"></symbolconfigobject></pre>	
7	-	
8	<pre>TypeList></pre>	
9	<typesimple iecname="BYTE" name="T_BYTE" size="1" swapsize="1" typeclass="Byte"></typesimple>	
10	-	
11	Image: State of the state o	
12	A <node name="eCoV3_0PCDA"></node>	
13	<pre><node name="GVL_0PCDA"></node></pre>	
14	<pre><node access="ReadWrite" name="GVL_Input" type="T_BYTE"></node></pre>	
15	<pre><node access="ReadWrite" name="GVL Output" type="T BYTE"></node></pre>	
16	-	
17	-	
18	-	
19	<pre>L</pre>	

7.2 CP600

When changing the PLC from AC500 V2 to AC500 V3 the protocol and the symbol file or addresses are changed. In this chapter it is explained, how to update the Panel Builder project to establish the communication with the existing project to the AC500 V3 PLC.



CAUTION!

Before starting the conversion of the tags from AC500 V2 protocol to AC500 V3 protocol a copy of the Panel Builder / Automation Builder project should be saved as backup.



There are two possibilities how to change from an AC500 V2 protocol to an AC500 V3 protocol.

- 1. Renaming the used tags from AC500 V2 style to AC500 V3 style
- 2. Replacing the AC500 V2 tags by AC500 V3 tags in the widgets

The Renaming of the tags, like described in chapter 7.2.1, is only working if the V2 tags have a path like in ABB CoDeSys ETH protocol. If there is no path only the replacing is working. The renaming should be used whenever there are many tags used in the project.

In case only a few tags are used in the project the recommendation is to replace the tags like described in chapter 7.2.2

The following two chapters describe the steps to rename or replace the existing variables. This is described exemplarily on the basis of ABB CoDeSys ETH protocol and ABB Modbus TCP protocol. The described steps are also working with all other protocols.

7.2.1 Renaming tags e.g. CoDeSys V2 ETH to CODESYS V3 ETH protocol

- 1. Update the project from AC500 V2 PLC to AC500 V3 PLC as described in chapter 5
- 2. If not already done, create a symbol file as described in chapter 7.1.2
- 3. If the Panel Builder project is inside the Automation Builder project
 - a) Open Panel_CP600 from the devices tree
 - b) Make sure the AC500 V3 PLC connection is checked

Project	Panel Builde	r Project Information			
	Project name:	Panel Project			
Information	Panel type: Panel size (piz Panel orientat	CP6607 el): 800 x 480 on: Landscape			
	Number of pa	ges: 0			
	Last edited:	7/28/2022 10:13 /	M		
	Description				
	Please enter	our description.			
	Connection				
	Con	nect PLC	Use Standard Conn. Settings	Connection Type	Communication Settings
		PLC_AC500_V2	\checkmark	Gateway 🔹	192.168.0.11:1200, 'ABB Tcp/Ip Level 2 AC', 'AC500 Default TCP-IP_'
	5	PM5630_2ETH		Gateway 🔻	192.168.0.10:11740, 'Gateway-1', TCP/IP

c) Check Update Panel Builder project on launch and click Launch Panel Builder Editor

Update Panel Builder project on launch

Launch Panel Builder Editor

- 4. If the Panel Builder project is separate from the Automation Builder project
 - a) Open Panel Builder
 - b) Go to Protocols and click 🕇
 - c) In the drop down list in column PLC select CODESYS V3 ETH

×J

- d) Set the IP address and confirm with OK
- e) Go to Tags
- f) Select CODESYS V3 ETH protocol
- g) Click Import Dictionary
- h) Select CODESYS3 xml Hierarchical
- i) Browse to the symbol file saved next to the Automation Builder project
- j) Select the symbol file and click open
- 5. Go to the Tags in Panel Builder. Here the old ABB CoDeSys ETH is inside as well as the new CODESYS V3 ETH. In both protocols the same tag names are existing. But the path to these tags is different.

AC500 V2 Tags		
arrStData[9].bStatus	UINT	PLC_AC500_V2//arrStData[9]/bStatus
arrStData[9].rValue	REAL	PLC_AC500_V2//arrStData[9]/rValue
arrStData[9].xStart	BOOL	PLC_AC500_V2//arrStData[9]/xStart
strMessage	STRING	PLC_AC500_V2//strMessage
xReadData	BOOL	PLC_AC500_V2//xReadData
— HMI_PRO.iRetainedVar	INT	PLC_AC500_V2/HMI_PRO/iRetainedVar
— HMI_PRO.sRetainedString	STRING	PLC_AC500_V2/HMI_PRO/sRetainedString
- PLC_PRG.lrVar	LREAL	PLC_AC500_V2/PLC_PRG/lrVar
PLC_PRG.udiVar	UDINT	PLC_AC500_V2/PLC_PRG/udiVar

AC500 V3 Tags

hj

▲ [9]	Container	
- bStatus	UINT	PM5630_2ETH/Application/Global_Variables/arrStData[9]/bStatus
- rValue	REAL	PM5630_2ETH/Application/Global_Variables/arrStData[9]/rValue
- xStart	BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[9]/xStart
- strMessage	STRING	PM5630_2ETH/Application/Global_Variables/strMessage
- xReadData	BOOL	PM5630_2ETH/Application/Global_Variables/xReadData
# HMI_PRO	Container	
- iRetainedVar	INT	PM5630_2ETH/Application/HMI_PRO/iRetainedVar
- sRetainedString	STRING	PM5630_2ETH/Application/HMI_PRO/sRetainedString
PLC_PRG	Container	
- Ir Var	LREAL	PM5630_2ETH/Application/PLC_PRG/lrVar
- udiVar	UDINT	PM5630_2ETH/Application/PLC_PRG/udiVar

The following description is focusing on the update where the AC500 V2 PLC, the AC500 V3 PLC and the Panel Builder project are all in the same Automation Builder project.

If they are not in the same project and no alias names are set the PLC names might be missing. The following steps can be done 1 to 1 just the names are different

- Rename global variables
 V2 Global variables start with PLC_AC500_V2//
 V3 Global variables start with PM5630_2ETH/Application/Global_Variables/
 - a) Select all tags in the AC500 V2 protocol which are global

b) Click Find and Replace from the menu G >] [> \$B B> 하 ₽- Search Filter by: E R hì Туре Data Tag name Find and Replace ABB CoDeSys ETH:prot1 ۵ Alias: PLC_AC500_V2 Container Model: Motorola Application Container arrStData[0].bStatus PLC_AC500_V2//arrStData[0]/bStatus UINT arrStData[0].rValue REAL PLC_AC500_V2//arrStData[0]/rValue PLC_AC500_V2//arrStData[0]/xStart .arrStData[0].xStart PLC_AC500_V2//arrStData[1]/bStatus c) Confirm with Yes d) Find what: PLC AC500 V2// e) Rename with: PM5630_2ETH/Application/Global_Variables/ N The expressions above have to be adapted to the naming used in the pro-(御) ject f) Click Rename Selected 🔚 Tag Find and Rename × Column: Name Rename Names in Dictionary Find what: PLC_AC500_V2// Rename with: PM5630_2ETH/Application/Global_Variables/

Case sensitive	Rename Selected Rename All Cancel
----------------	-----------------------------------

7. Repeat with the local variables

Find what: PLC_AC500_V2/

Rename with: PM5630_2ETH/Application/

- 8. Move AC500 V2 variables to AC500 V3
 - a) Please recheck that all variables below ABB CoDeSys ETH have the same names as in CODESYS V3 ETH
 - b) Delete the V3 Dictionary

	- Occarry			
	necipe			F
~	Contractionaries			
	🚺 [ABB CoDeSys ETH prot1] Application			F
	[ABB CoDeSys ETH V3 prot2] Application			. 1
>	🗁 Keypads	<u>A</u>	Rename dictionary	E
		-	Delete dictionary	E
		_	2	

All tabs below CODESYS V3 ETH are gone now

c) Select all AC500 V2 tags in ABB CoDeSys ETH

d) In the properties window on the right side. Change the protocol from ABB CoDeSys ETH to CODESYS V3

ata	Type	Tag name	Property	Value
ABB CoDeSys ETH:prot1			Y Driver	
Alias: PLC_AC500_V2 Model: Motorola	Containe		Alias	PLC_AC500_V2
 Application 	Containe		Model	Motorola
.arrStData[0].bStatus	UINT	PM5630_2ETH/Application/Global_Variables/arrStData[0]/bStatus	Protocol	ABB CoDeSys ETH:prot1
.arrStData[0].rValue	REAL	PM5630_2ETH/Application/Global_Variables/arrStData[0]/rValue	✓ Tag	ABB CoDeSys ETH:prot1
.arrStData[0].xStart	BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[0]/xStart	Data Type	CODESYS V3 ETH:prot2
.arrStData[1].bStatus	UINT	PM5630_2ETH/Application/Global_Variables/arrStData[1]/bStatus	Tag name	<multiple selection=""></multiple>
.arrStData[1].rValue	REAL	PM5630_2ETH/Application/Global_Variables/arrStData[1]/rValue	PLC tag name	<multiple selection=""></multiple>
.arrStData[1].xStart	BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[1]/xStart	Groups	
.arrStData[2].bStatus	UINT	PM5630_2ETH/Application/Global_Variables/arrStData[2]/bStatus	Tag URI	<multiple selection=""></multiple>
.arrStData[2].rValue	REAL	PM5630_2ETH/Application/Global_Variables/arrStData[2]/rValue	Comment	sharapic selections
.arrStData[2].xStart	BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[2]/xStart	Rate	500
.arrStData[3].bStatus	UINT	PM5630_2ETH/Application/Global_Variables/arrStData[3]/bStatus		
.arrStData[3].rValue	REAL	PM5630_2ETH/Application/Global_Variables/arrStData[3]/rValue	R/W	R/W
.arrStData[3].xStart	BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[3]/xStart	Active	false
.arrStData[4].bStatus	UINT	PM5630_2ETH/Application/Global_Variables/arrStData[4]/bStatus	Simulator	Variables
.arrStData[4].rValue .arrStData[4].xStart	REAL BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[4]/rValue PM5630_2ETH/Application/Global_Variables/arrStData[4]/xStart	Scaling	None
.arrStData[5].bStatus	UINT	PM5630_2ETH/Application/Global_Variables/artstData[4]/Xstart PM5630_2ETH/Application/Global_Variables/artStData[5]/bStatus	Min value	
.arrStData[5].rValue	REAL	PM5630_2ETH/Application/Global_Variables/arrStData[5]/rValue	Max value	
.arrStData[5].xStart	BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[5]/xStart	> Dictionary	
arrStData[6] hStatus	LITNET	DMEE20, DETH/Application/Clobal_Variables/arrotata[6]/Actatus		

- e) Acknowledge the pop up window with OK
 All tags below ABB CoDeSys ETH are now grey
 The tags can be found in CODESYS V3 ETH now
- f) Recheck in any Page, that the PLC variables are still mapped correct
- 9. Reload symbol file using Panel Builder in an Automation Builder project
 - a) Save the Panel Builder project
 - b) Close the Panel Builder project
 - c) Uncheck the Connect box for PLC_AC500_V2

Please	e enter your	description.					
Conne	ections						
	Connect	PLC	Use Standard Conn. Settings	Connection Typ	be	Communication Settings	Details
•		PLC_AC500_V2		Gateway	•	192.168.0.11:1200, 'ABB Tcp/Ip Level 2	
		PM5630_2ETH	\checkmark	Gateway	•	192.168.0.10:11740, 'Gateway-1', TCP/IP	
						Update Panel Builder project on	launch
						Launch Panel Builder E	ditor
						Edditer Farler Balder E	arcor

d) Check the Box Update Panel Builder project on launch and click Launch Panel Builder Editor

10. Reload Symbol File using Panel Builder as standalone

- a) Click import dictionary
- b) Open the symbol file from the file system
- 11. Update tags
 - a) In the Tags view now several grey entries can be found

b) Select them all and click Import Tags

+ - 2 0 0 3 0	♣ ■ ★ R P- Search	Tilter by: Data	
Data	Import Tags	Tag name	
— Application/PLC_PRG/IrVar		PM5630_2ETH/Application/PLC_PRG/IrVar	
Application/PLC_PRG/udiVar	BOOL	PM5630_2ETH/Application/PLC_PRG/udiVar	
Application	Container		
Global_Variables	Container		
arrStData	Array[10] : MachineParameters	Application/Global_Variables/arrStData	
⊿ [0]	Container		
bStatus		PM5630_2ETH/Application/Global_Variables/arrStData[0]/bSt	tatus
- rValue	REAL	PM5630_2ETH/Application/Global_Variables/arrStData[0]/rVal	lue
xStart	BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[0]/xSt	
⊿ [1]	Container		
- bStatus	UINT	PM5630_2ETH/Application/Global_Variables/arrStData[1]/bSt	tatus
rValue	REAL	PM5630_2ETH/Application/Global_Variables/arrStData[1]/rVal	lue
xStart	BOOL	PM5630_2ETH/Application/Global_Variables/arrStData[1]/xSt	
▲ [2]	Container		
- bStatus	UINT	PM5630_2ETH/Application/Global_Variables/arrStData[2]/bSt	tatus

- c) Confirm with Yes to All
- d) Check Tag Cross Reference and show Invalid Tag Reference

5.	Tag Cross Reference*			+	×
🔆 ObjectView	Group by: Tag 🔹	Show:	Invalid Tag Referen All References	ce	•
ectVie	₽- Search	7	Invalid Tag Referen	ice	
×					_
~					
Tag					
Cross					
X Tag Cross Reference*					
rence					
*					

No entry should be in the list

- 12. Download the updated panel builder project to the CP600 and make sure it is communicating with the AC500 V3 PLC
- 13. Delete the ABB CoDeSys ETH protocol from the Protocols and download it again to the CP600

1:Page1 Tags Protocols x		
+ - ^ ~ 5		
PLC Delete Protocol	Configuration	Dictionaries
ABB CODOUS Emploi	CfgVer=1 prefix=PLC_AC500_V2 ip_Address=192.168.0.11 Port=1200 Block Size=1430 Timeout	[ABB CoDeSys ETH prot1] Application
CODESYS V3 ETH:prot2	CfgVer=1 prefix=PM5630_2ETH prot=TCP ipAddress=192.168.0.10 timeout=1000 maxVarList=2	[ABB CoDeSys ETH V3 prot2] Application

7.2.2 Replacing tags e.g. ABB Modbus TCP V2 to ABB Modbus TCP V3

The workaround described in chapter 7.2.1 is only working with limitations for ABB Modbus. The steps can be done 1 to 1 as described for CODESYS ETH protocol. But the renaming of the tags is not possible, as there is no prefix like in the CODESYS ETH protocol. If following the steps 6 and 7, described in the last chapter, the adding of the path must be done for each tag by hand.

The recommended possibility is to not modify the tags by renaming but replace all used variables from AC500 V2 protocol to AC500 V3 protocol. In the description below the required steps for replacing the variables are described.

1. Add the Modbus TCP protocol and the tags via the export file to the project

ABB Modbus TCP:prot1		CfgVer=1 port=502. timeout=1000. UnitID=1 model=4	[ABB Modbus TCP prot1] MODBUS
ABB Modbus TCP:prot2		CfgVer=1 port=502. timeout=1000. UnitID=1 model=8	[ABB Modbus TCP prot2] PanelUpgradeV2V
ource: 🕑 Tag 🔾 Alias 🔾 System	n 🔾 Widget 🔾	Recipe	
₽- Search	Tilte	er by: Data 🔹 Protocol: Show all 👻 🗹 Show all tags	
Data	Туре	Tag name	Pro
 ABB Modbus TCP:prot1 Model: ABB PM59x, PM58x, PM573 Ethe 	rnet Container		×
arrStData[0]	unsignedByte	arrStData[0]	
arrStData[1]	unsignedByte	arrStData[1]	
arrStData[2]	unsignedByte	arrStData[2]	×
arrStData[3]	unsignedByte	arrStData[3]	
- iRetainedVar	short	iRetainedVar	
- Ir Var	double	IrVar	
- sRetainedString	string-80	sRetainedString	
strMessage	string-80	strMessage	
- udiVar	unsignedInt	udiVar	
xReadData	boolean	xReadData	
ABB Modbus TCP:prot2 Model: ABB PM56xx Ethernet	Container		
Global Variables/arrStData[0]	unsignedByte	Global Variables/arrStData[0]	
Global Variables/arrStData[1]	unsignedByte	Global Variables/arrStData[1]	
Global Variables/arrStData[2]		Global Variables/arrStData[2]	
Global Variables/arrStData[3]	unsignedByte	Global_Variables/arrStData[3]	
Global_Variables/strMessage	string-80	Global_Variables/strMessage	
Global_Variables/xReadData	boolean	Global_Variables/xReadData	
	short	HMI_PRO/RetainedVar	
HMI_PRO/sRetainedString	string-80	HMI_PRO/sRetainedString	
PLC_PRG/IrVar	double	PLC_PRG/rVar	
PLC_PRG/udiVar	unsignedInt	PLC PRG/udiVar	

2. Open Tag Cross Reference View, pin it and click refresh view at the bottom

ferences 🗸
r by: Tag 🔹

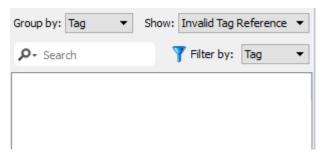
3. Select the first tag in the cross reference list and click Replace

>	arrStData[0]	Replace
	arrC+Data[1]	6

4. Click Select new tag and choose the corresponding V3 Tag from the dialog and confirm with OK

		P- Search	7 Filte	r by: Data 👻 Protoco	show all 👻	Show all tags
arrStData[0]	Select new tag Cance	Data	Type	Tao name		
arrStData[1]			Type	Tay hame		
arrStData[2]		ABB Modbus TCP:prot1 Model: ABB PM59x, PM58x, PM573 Ethernet	Container			
arrStData[3]		arrStData[0]	unsignedByte	arrStData[0]		
		arrStData[1]	unsignedByte			
RetainedVar		arrStData[2]	unsignedByte			
IrVar		arrStData[3]	unsignedByte	arrStData[3]		
sRetainedString		RetainedVar	short	iRetainedVar		
		- IrVar	double	IrVar		
strMessage		sRetainedString	string-80	sRetainedString		
udiVar		strMessage	string-80	strMessage		
xReadData		udiVar	unsignedInt	udiVar		
		xReadData	boolean	xReadData		
		ABB Modbus TCP:prot2 Model: ABB PM56xx Ethernet	Container			
		Global_Variables/arrStData[0]		Global_Variables/arrStData[0]		
		Global_Variables/arrStData[1]		Global_Variables/arrStData[1]		
		Global_Variables/arrStData[2]		Global_Variables/arrStData[2]		
		Global_Variables/arrStData[3]		Global_Variables/arrStData[3]		
		Global_Variables/strMessage	string-80	Global_Variables/strMessage		
		Global_Variables/xReadData	boolean	Global_Variables/xReadData		
		HMI_PRO/RetainedVar	short	HMI_PRO/iRetainedVar		
		HMI_PRO/sRetainedString	string-80	HMI_PRO/sRetainedString		
		PLC_PRG/IrVar	double	PLC_PRG/irVar		
		PLC_PRG/udiVar	unsignedInt	PLC_PRG/udiVar		

- 5. Repeat the last two steps above to replace all AC500 V2 tags by AC500 V3 tags
- 6. Make sure that all variables in the Tag Cross Reference (Show = All References) are AC500 V3 variables
- 7. Delete the AC500 V2 protocol from Protocols
- 8. Change the Cross Reference View from "All References" to "Invalid Tag Reference" and make sure, that there is no entry in the list



9. Hide the cross reference view again, download the project to the HMI and ensure that the communication with the AC500 V3 PLC is running as expected.

Tag Cross	Reference			1	×	Projec
Group by:	Тад	•	Show:	Invalid Tag Reference	Auto	Hide

7.3 OPC DA

This chapter gives an introduction of the steps for the use of OPC DA with AC500 V2 and AC500 V3:

- OPC DA for AC500 V2, e.g. PM566-TP-ETH
- OPC DA for AC500 V3, e.g. PM5072-T-2ETH
- Update the project from AC500 V2 to AC500 V3

CAUTION!

Before starting the conversion of the OPC DA from V2 to V3 a copy of the Panel Builder / Automation Builder project should be saved as backup.

7.3.1 OPC DA for AC500 V2

Install the OPC server AC500 V3 or AC500 V2 as additional tools with the Automation Builder installer.

Install Software Packages for:	Description	Release Notes	Packages		
 Automation Builder Standalone Tools ABB License Manager 		CODES	YS OPC Server 3		*
AC500 - Tools AC500 - Tools Cobesys OPC Server 3.5 Cobesys OPC Server 2.3 P Configuration Tool MultiOnlineChange Tool	E D U N D			s the IEC 61131-3 process data .E for process control)	8
HMI Panel Builder 600 Runtime for PC Control Panel - CP600	This soft and featu		ains the following m	ain components	
= Safety PLC - Pluto		client support and n connection to AC50			

Note: Prerequisite for the OPC Server configuration is that a symbol file is configured and downloaded to the AC500 V2 PLC as described in chapter AC500 V2 PLC7.1.1.

7.3.1.1 Configure OPC Server

(hij)

Since OPC Server AC500 V2 and AC500 V3 can be installed in parallel on the same PC, we will use OPC server V3 in this example which can support OPC connection to AC500 V2 and AC500 V3 PLCs.

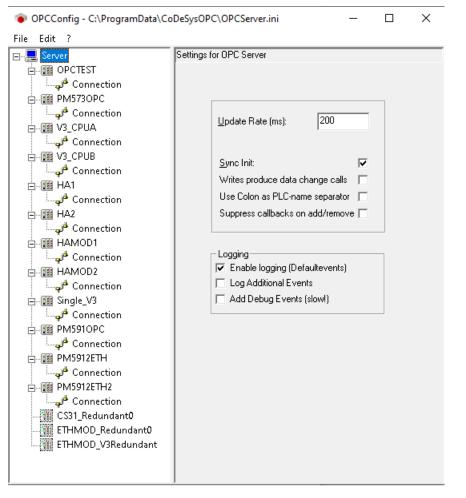
Start via 3S CODESYS/CoDeSysOPC Server V3/OPC Configurator, or start the OPC configurator by fold-er C:\Program Files (x86)\3S CODESYS\CODESYS OPC Server 3\ OPCConfig.exe directly.

 3S CODESYS CODESYS Gateway V3 CODESYS OPC Server V3 AE Configurator User Guide AE Configurator CODESYS OPC Server V3 Benutzer CODESYS OPC Server V3 User Guide
OPC Configurator
Gateway V2.3
3S Software
ABB
ABB Automation Products GmbH
ABB Local Applications
ABB Standard Applications
Accessories
Audio Related Programs
📙 Business Explorer 🛛 👻
◀ Back
Search amorane and files
Search programs and files 🔎

When "OPCConfig.exe" is called, the current configuration of the "OPCServer.ini" is displayed.

OPCConfig - C:\Progra	amData\CoDeSysOPC\OPCServer.	ini Marcalle Constantin			
File Edit ?					
Server		94			
Connectio	Save As	and Dis			X
Connectio	Computer	Local Disk (C:) ProgramData CoDeSysOPC	• \$j	Search CoDeSysOPC	٩
Connectio	Organize 🔻 New folder			8==	• 🕡
	★ Favorites	Name	Date modified	Туре	Size
	E Desktop E Downloads Recent Places	E OPCServer.ini	06.03.2017 16:42	Configuration sett	2 KB
		DPCServer_HA.ini	02.08.2016 12:00	Configuration sett	2 KB
		OPCServer_HA1.ini	16.09.2016 14:11	Configuration sett	2 KB
	OneDrive - ABB OneDrive - Personal	OPCServer11.ini	14.09.2016 09:19	Configuration sett	1 KB
	☐ Libraries ☐ Documents ♪ Music				
		* <			•
	File name: OPCSer Save as type: OPCCo	verini nfig Files (*.ini)			- -
	Alide Folders			Save	Cancel

If the configuration is required for other projects, save it under a new name.



Append a new PLC under Server tree, and make the settings as shown below:

⊡ <mark></mark> Server	Settings for eCoV2		
i⊟…∰ eCoV2 iar Connection	<u>I</u> nterface: Project <u>n</u> ame:	GATEWAY	<u> </u>
		<u>⊺</u> imeout (ms): Number of <u>⊺</u> ries:	3
		<u>B</u> uffer Size (Byte):	4800
		<u>W</u> ait Time (s): <u>R</u> econnect Time (s):	10 15
	Active: Motorola Byteorder: No Login-Service: ▼		
	Logging ✓ Enable logging (Defau □ Log Additional Events □ Add Debug Events (sl		

The project name is not required because the symbol information should be uploaded from AC500-eCo V2.

Open the connection settings and choose a channel of the channel list (normally the channel which is used for programming), or click button 'New...' to create a new channel.

Communication Parame	ters	\times
Channels □- Local 	Tcp/lp Name Value Address 192.168.2.10 Port 1201	OK Cancel
	Motorola byteorder Yes Communication Parameters: New Channel Name eCo_V2 Device	OK Cancel
	Name Info ABB Arcnet AC ABB SST Arcnet AC ABB RS232 AC ABB RS232 AC drive ABB Tcp/Ip Level 2 AC ABB SST Tcp/Ip Le Tcp/Ip 35 Tcp/Ip driver Serial (RS232) 35 Serial RS232 drive	

If more than one PLC, then repeat for the other PLCs. After that save as 'OPCServer.ini' and close the OPCConfig tool.

7.3.1.2 Check OPC Server with MatrikonOPC Explorer.

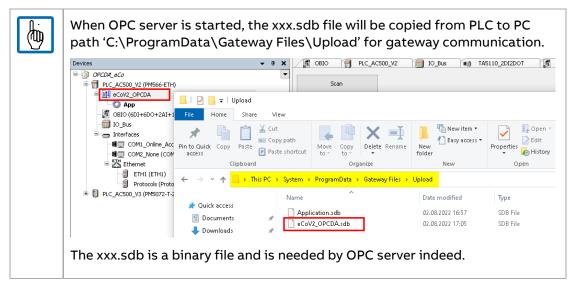
- Start MatrikonOPC Explorer
- Connect CoDesys.OPC.DA
- Add Group
- Add Items → Select available Items in 'Server CoDeSys.OPC.DA'
- Add to Tag List and close the Item browser

📥 MatrikonOPC Explorer (\$Group_1)	? ×	
File Edit View Browse		
File Edit View Browse Image: Second Sec	Tags to be added:	X
Available Items in Server 'CoDeSys.OPC.DA':	Tag: eCoV2GVL_Input Access Path: Validated: No Data Type: Empty/Default Create Active: Yes Readable: ??? Writeable: ???	

Now the variables will be shown and automatically updated.

🥺 MatrikonOPC Explorer - [Untitled	i*]					
File Server Group Item View	Help					
윤 🗶 🖀 🙆 🔯 🕹	🔒 📝 🕮 🎂 💕 🗰	(🍅 🛛	4			
\$Group_1	Contents of '\$Group_1'					
戸…😼 Localhost '\\DE-L-7268277'	Item ID	Acce	Value	Quality	Timestamp	Status
ABB.DriveDA.1	eCoV2GVL_Input		51	Good, non-specific	08.02.2022 4:59:40.080 PM	Active
CoDeSys.OPC.02	eCoV2GVL_Output		15	Good, non-specific	08.02.2022 4:59:50.159 PM	Active
CoDeSys.OPC.AE.1						
E CoDeSys.OPC.DA						
\$Group_1						
Matrikon.OPC.Simulation.1						
🗄 🔎 Notwork Noisbborbood						

If the configuration is correct, then CoDeSys.OPC.DA is connected. The OPC Client is running and the quality of the items is good. The values of the items can be read and written by an OPC client.



7.3.2 OPC DA for AC500 V3

Note: Prerequisite for the OPC Server configuration is that a symbol file is configured and downloaded to the AC500 V3 PLC as described in chapter 7.1.2.

7.3.2.1 Configure OPC Server

Start via 3S CODESYS/CoDeSysOPC Server V3/OPC Configurator, or start the OPC configurator by fold-er C:\Program Files (x86)\3S CODESYS\CODESYS OPC Server 3\ OPCConfig.exe directly.

OPCConfig - C:\ProgramDa	ata\CoDeSysOPC\OPCServer.ini – 🗆 🗙
File Edit ?	
Erver È	Settings for OPC Server
	Update Rate (ms): 200
	Sync Init:
	Writes produce data change calls
	Use Colon as PLC-name separator 🛛
	Suppress callbacks on add/remove
	Logging Enable logging (Defaultevents)
	Log Additional Events
	Add Debug Events (slow!)

Append a new PLC under Server and make the settings as shown below:

🐞 OPCConfig - C:\ProgramDat	ta\CoDeSysOPC\OPCServer.ini — 🗆 🗙
File Edit ?	
🖃 📃 Server	Settings for eCoV3
⊡ ب∰ eCoV2 می الله Connection ⊡ ب∰ eCoV3 می الله Connection	Interface: GATEWAY3
	<u>I</u> imeout (ms): 10000
	Number of <u>I</u> ries: 3
	<u>B</u> uffer Size (Byte):
	<u>₩</u> ait Time (s): 10
	<u>R</u> econnect Time (s): 15
	Active: Image: mail of the second secon
	Logging ✓ Enable logging (Defaultevents) ─ Log Additional Events ─ Add Debug Events (slow!)

Open the Codesys address of PLC dialog by pushing Edit button, then enter the PLC IP address and Port.

🐞 OPCConfig - C:\Prog	ramData\CoDeSysOPC\OPCServer.ini		- 🗆 ×
File Edit ?			
⊡… <mark>,</mark> Server	Settings for connection to eCoV3		
ianiania eCoV2 internation internation eC <u>oV3</u>	Edit		Expert
<u>در منابع منابع المنابع المنابع</u>	Gateway: TCP/IP	Address: k Port: 1	
	CoDeSys address of PLC	×	
	Gateway IP address localhost Port ,1217	ОК	Device Scan
	PLC name or address	0000	
	Use TCP/IP blockdriver	v	
	IP address of PLC	192.168.2.11	
	IP port of PLC	11740	

After closing the dialog with OK the settings will be displayed in the main dialog.

OPCConfig - C:\ProgramD	ata\CoDeSysOPC\OPCServer.ini	_		\times
File Edit ?				
🖃 💂 Server	Settings for connection to eCoV3			
eCoV2	Edit	Expert		
ian <mark>- ∭</mark> eCoV3 اسم ب ^{اله} Connection	Colourse TCD//D			
	Gateway: TCP/IP Address: localhost Port: 1217			
	Device:			
	Parameter Value Comment IpAddress 192.168.2.11 Address of CoDeSys V3 Device		1	

If more than one PLC, then repeat for the other PLCs.

Save as by default "OPCServer.ini" in the installation directory and close the OPCConfig tool..

OPCConfi	g - C:\ProgramData\	CoDeSysO	PC\OPCServer.ini	—	1 ×
File Edit ?					
⊡ <mark></mark> _ Server		ettings for co	onnection to eCoV3		
eCo	Connection		Edit	Expert	
	🐞 Save As	C	atoman TCD/ID Address Issue		×
	$\leftarrow \ \ \rightarrow \ \ \checkmark \ $	→ ✓ ↑ 🔄 ≪ Progr → CoDeSysOPC 🗸 Ö 🔎 Search CoDeSysOPC			
	Organize 🔻	New folde	er	÷== •==	• • •
	🛖 DEU21961	9\$ (\\E 🔨	Name	Date modified	Туре
	🛖 Developm	ientRel	🐻 OPCServer.ini	02.08.2022 17:05	Configure
	🛖 Developm	ientToc	📓 OPCServer_default.ini	20.04.2012 11:59	Configura
	🛖 Integratio	nTestB	📓 OPCServer_Example.ini	02.08.2022 16:19	Configura
	🛖 PGPLCDat	taExch			
	🛖 ReleaseVer	rsions			
	🏪 System	¥	٢		>
	File nan	ne: OPCS	erver.ini		~
	Save as ty	pe: "OPCO	Config Files (*.ini)		~
	∧ Hide Folders		[Save Car	ncel .::

7.3.2.2 Check OPC Server with MatrikonOPC Explorer.

- Start MatrikonOPC Explorer
- Connect CoDesys.OPC.DA
- Add Group
- Add Items \rightarrow Select available Items in 'Server CoDeSys.OPC.DA'
- Add to Tag List and Close the Item browser

AntrikonOPC Explorer (\$Group_1)	? ×	
File Edit View Browse Image: Second Sec	o be added:	
Filter: Data Type Filter: Empty/Default ✓ Write Access @ Read Access Branches Available Items in Server 'CoDeSys.OPC.DA': Image: Proceeding of the ecoval		
Image: Solution of Concernance Image: Solution of Conce Image: Solution of Concernanc		×

Now the variables will be shown and automatically updated.

🥺 MatrikonOPC Explorer - [Untit	tled*]					
File Server Group Item Vie	w Help					
윤 🕷 🐨 🖻 🍲 🔗 🎽	(🔒 📝 🛤 📥 🛥 🗮 🍑 😭					
\$Group_1	Contents of '\$Group_1'					
	Item ID	A	Value	Quality	Timestamp	Status
ABB.DriveDA.1	eCoV3.eCoV3_OPCDA.GVL_OPCDA.GVL_Input		0	Good, non-specific	08.03.2022 2:30:40.723 PM	Active
CoDeSys.OPC.02 GC CoDeSys.OPC.02 GC CoDeSys.OPC.AE.1 G- CoDeSys.OPC.DA Goup_1 GC Matrikon.OPC.Simulation.:	BeCoV3.eCoV3_OPCDA.GVL_OPCDA.GVL_Output		15	Good, non-specific	08.03.2022 2:30:50.659 PM	Active

If the configuration is correct, then CoDeSys.OPC.DA is connected. The OPC Client is running and the quality of the items is good. The values of the items can be read and written by an OPC client.

7.3.3 Update the project from AC500 V2 to AC500 V3

If the user has an old project of AC500 V2, and now wants to upgrade to an AC500 V3 PLC, its OPCDA communication settings must be reconfigured. For details, please refer to the above sub chapter 7.3.2. From the user's point of view, its configuration operation will become simpler compared to the AC500 V2 OPC DA.

7.4 UDP

For AC500 V3 PLC, UDP is no longer configurable in the device tree. A UDP communication can be established with the library **Net Base Services** from Codesys.

An application example is currently in preparation. If a pre version is required, please contact plc.support@de.abb.com.

7.5 MQTT & JSON

An MQTT application using also the JSON library can be converted to V3 quite simple. As both MQTT and JSON library in V2 are already in the PLC open style no adaptions for the function block calls are necessary. If constants are used, they need to be accessed via the namespace in V3.

E.g. MQTT_MAX_TOPIC_LEN needs to be accessed via the namespace AC500_MQTT.Constants.MQTT_MAX_TOPIC_LEN

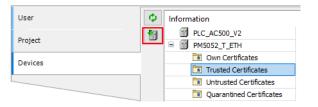
ReceivedTopic : STRING (AC500_MQTT.Constants.MQTT_MAX_TOPIC_LEN);

For saving the error numbers it is recommended to not use WORD variables but use variable type AC500_MQTT.ERROR_ID.

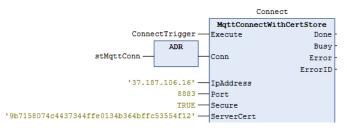
ConnectLastErrorID : AC500_MQTT.ERROR_ID;

The AC500 V3 PLCs offer the possibility to use a certification store.

In AC500 V2 client and server certificates could only be stored as files in the filesystem or used as string buffer. These possibilities are also available for AC500 V3 but the recommendation is to use the inbuild Certificate Manager accessible via the security screen. There the CA certificate can be downloaded to the Trusted Certificates.



For more information regarding Security please check chapter 2.6. The MQTT library in AC500 V3 offers the possibility to use the function block MqttConnectWithCertStore for using a certificate from the cert store. The Thumbprint of the certificate needs to be used as input for this function block.



Similar like described in the steps above also the code for the JSON library can be updated.

 JSON_MAX_KEY_LEN must be replace by AC500_JSON.Constants.JSON_MAX_KEY_LEN • JSON_ERROR_ID must be replaced by AC500_JSON.ERROR_ID

After performing the steps above, you should be able to compile the MQTT/JSON part of the project again. The same functionality as in AC500 V2 is then available in AC500 V3.



ABB AG

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© 2023 ABB. All rights reserved