

# Quickstart Guide

## ABB PLC and drives integration using Modbus RTU



## Contents

Introduction.....	4
Safety instructions.....	4
Limitations .....	4
Fieldbus grounding.....	6
Fieldbus shield .....	6
End termination.....	6
Pull up/pull down resistors.....	6
Drives with embedded fieldbus.....	7
Fieldbus modules for ABB drives.....	8
Common recommendations.....	8
ABB cable TK505.....	9
AC500 COM1 – Master at the bus line end.....	11
AC500 COM2 – Master at the bus line end.....	11
AC500-eCo COM1 – Master at the bus line end.....	11
AC500-eCo COM2 – Master at the bus line end.....	12
Drive configuration.....	13
Starting up ACS310 drives with embedded Modbus RTU.....	14
Starting up ACS355 drives with FMBA-01 Modbus RTU adapter.....	15
Starting up ACS800 drives with RMBA-01 Modbus RTU adapter.....	17
Starting up ACS850, ACQ810 drives with embedded Modbus RTU.....	18
Starting up ACS880 drives with FSCA-01 Modbus RTU adapter.....	19
Starting up ACSM1 drives with FSCA-01 Modbus RTU adapter.....	20
Install drives library .....	21
Control Builder Plus for PLC and Drives.....	22
Hardware configuration .....	22
IP configuration of CPU (for Ethernet connection between PC and PLC) .....	24
CODESYS .....	25
CODESYS tabs .....	25

---

Quickstart Guide  
ABB PLC and drives integration using Modbus RTU

---

Library Manager .....	26
PLC logic.....	27
Download program to PLC.....	32
Create boot project .....	33
Test the program .....	34
Read and write more data between PLC and drive (optional) .....	36
Visualizations .....	43
Add more drives (optional).....	46
Actualization rate .....	47
Other useful documentation .....	47
Revision .....	47

## Introduction

This guide will take you through the hardware installation and configuration of ABB ACS Drives and PLC in order to prepare for Modbus RTU control. ABB-specific ready-made function blocks and visualizations from the PS553-DRIVES library will be used for the control of the drives. While the guide is applicable to all AC500 PLCs and all Modbus RTU compatible ABB ACS drives, the example is built up with an AC500-eCo PLC and an ACS355 drive.

**Note!** *The material in this application might need to be adapted according to actual equipment and function before it is used. Testing of the equipment must always be performed by the responsible start-up person according to current legislation before the equipment is placed in service. ABB does not take any responsibility for possible damage caused by using the material in this application (shown examples, data, project tools, etc.).*

## Safety instructions

Follow all safety instructions delivered with the used components:

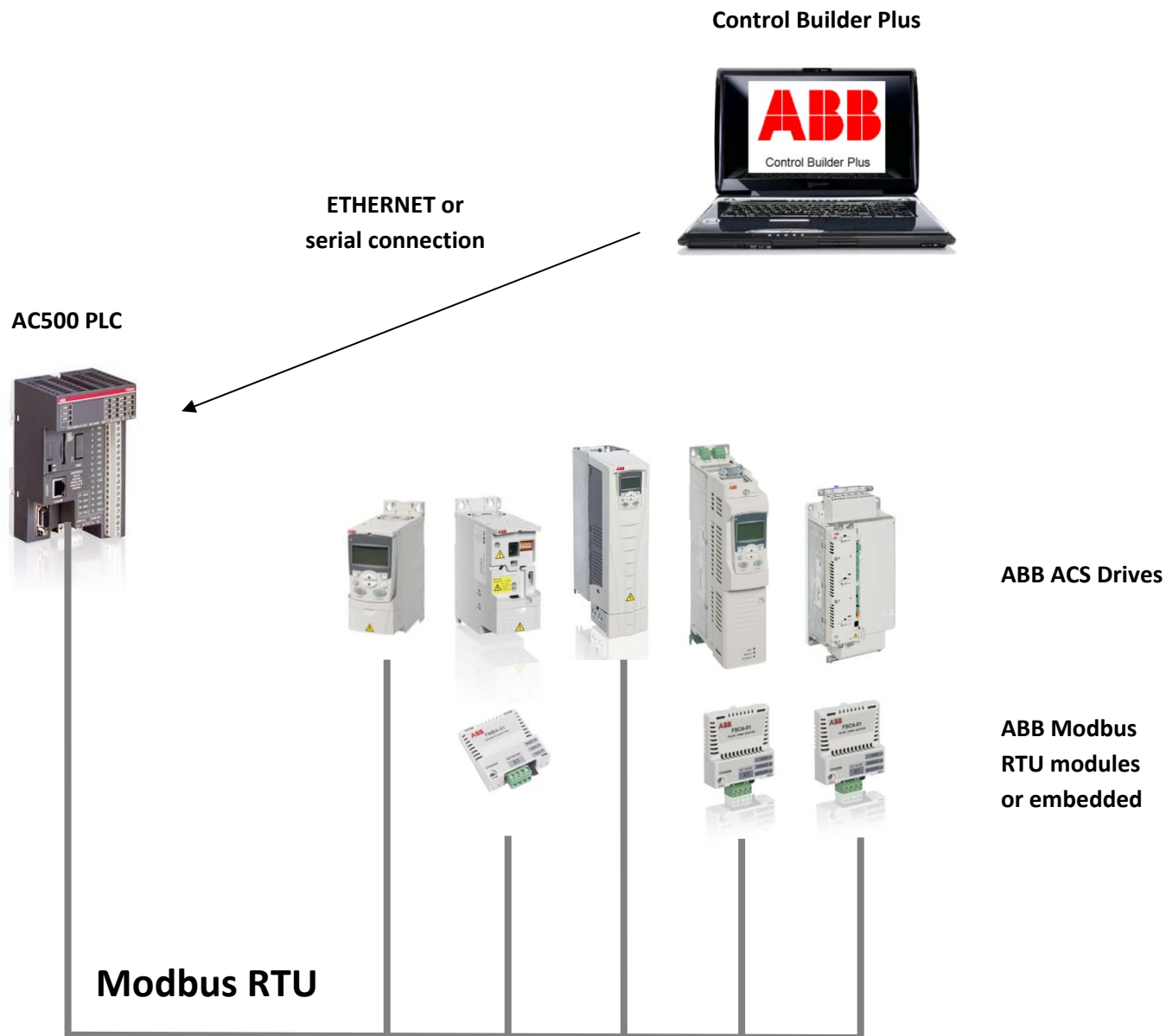
- Read the complete safety instructions for the used drive before you install, commission or use the drive. The complete safety instructions are given at the beginning of the drives user's manual (e.g. for ACS355: 3AUA0000066143 [English]).
- Read all safety and installation instructions of the used PLC. See the online help in CoDeSys (Help → Contents → Target System → AC500 / S500 → Introduction → Overview → Regulations).

## Limitations

The following limitations are valid for this guide:

- General
  - All instructions in this guide are based on default settings in PLC and drives
  - PS501 Control Builder Plus version 2.2.0 or later
  - The drive must be equipped with embedded Modbus RTU or a Modbus RTU adapter
- Function block and visualization library PS553-DRIVES compatibility
  - Communication profile: ABB drives
  - Drive types: All Modbus RTU compatible ABB ACS drives
  - Application types: Speed/Frequency control or Torque control

## Setup example



## Hardware physical connection

In the following chapter, physical connection of the hardware will be explained.

### Fieldbus grounding

For AC500, which has opto-isolated COM interfaces, it is recommended not to connect the ground. For AC500-eCo, where the COM interfaces are not opto-isolated, it is recommended to connect the ground at one place, typically in the bus master. For connection examples, see chapter “Installation examples”.

### Fieldbus shield

The shield should always be connected in a Modbus RTU line. The purpose of the shield is to protect against EMC disturbances and reflection on the bus line. For connection examples, see chapter “Installation examples”.

### End termination

- Purpose: To avoid signal reflections.
- Resistor size: 120 Ohms, ¼ W.
- Where to connect: One resistor in each end of the bus line (first and last node).

### Pull up/pull down resistors

- Purpose: To keep a defined high level on the bus while there is no data exchange.
- Resistor size: Depends on bus master (COM) voltage level. 470  $\Omega$  for AC500 (5 V output) and 270  $\Omega$  for AC500-eCo (3.3 V output).
- Where to connect: Activate pull-up and the pull-down resistors, but only once on every bus line, at the bus master.

## AC500 and AC500-eCo COM port overview

	AC500		AC500-eCo		CM574-RS
	COM1	COM2	COM1	COM2 (optional)	COM1=COM2 (see COM1 of AC500, except pull-up/pull-down)
COM Voltage level	5 V	5 V	3,3 V	3,3 V	5V
Opto-isolated (electrically isolated)	YES	YES	NO optional *)	NO	YES
Maximum bus cable length	Acc. to RS485 standard	Acc. to RS485 standard	50 m *) Acc to RS485 standard	50 m	Acc. to RS485 standard
Connection	Terminal	9 pole D-sub	9 pole D-sub *) Terminal	Terminal	Terminal
Termination resistor built-in	NO	NO	NO	NO	NO
Termination resistor size	120 $\Omega$	120 $\Omega$	120 $\Omega$	120 $\Omega$	120 $\Omega$
Pull-up/pull-down resistors built-in	YES (connect with bridges)	NO	NO *) yes connect with bridges	YES (connect with bridges)	YES (connect with bridges)
Pull-up/pull-down resistor size	470 $\Omega$	470 $\Omega$	270 $\Omega$	270 $\Omega$	475 $\Omega$
Pull-up/pull-down resistors connection point	Master (CPU) COM port	Master (CPU) COM port	Master (CPU) COM port	Master (CPU) COM port	Master (CPU) COM port
Pull-up/pull-down connection	Bridges: 1 to 2 and 3 to 4	External 470 $\Omega$ resistors	External 270 $\Omega$ resistors	Bridges: 1 to 2 and 3 to 4	Bridges: 1 to 2 and 3 to 4

\*) using TK506 RS485 isolator for PM554 / PM564 with index A3 and higher

## Drives with embedded fieldbus

The following ABB drives have Modbus RTU embedded:

- ACS310
- ACS/ACH550
- ACS850

Since the built-in end termination of the drive automatically also activates pull-up/pull-down resistors, it is recommended to skip the built-in option.

It is recommended to connect a separate end termination resistor to the first node (i.e. the PLC COM interface) and the last node (i.e. the last drive on the bus).

For connection examples, see chapter "Installation examples".

### Fieldbus modules for ABB drives

The following ABB drives Modbus RTU modules have an option to activate built-in end termination and pull-up/pull-down resistors.

- RMBA-01 (used with for instance ACS550 and ACS800)
- FSCA-01 (used with for instance ACS880 and ACSM1)

The following ABB drives Modbus RTU module does not have an option to activate built-in end resistors including pull-up/pull-down resistors.

- FMBA-01 (used with for instance ACS355)

Since the built-in end termination of the modules (RMBA-01 and FSCA-01) automatically also activates pull-up/pull-down resistors, it is recommended to skip the built-in option.

It is recommended to connect a separate end termination resistor to the first node (i.e. the PLC COM interface) and the last node (i.e. the last drive on the bus).

For connection examples, see chapter “Installation examples”.

### Common recommendations

**Note!** The following bullets are only recommendations and no general rules.

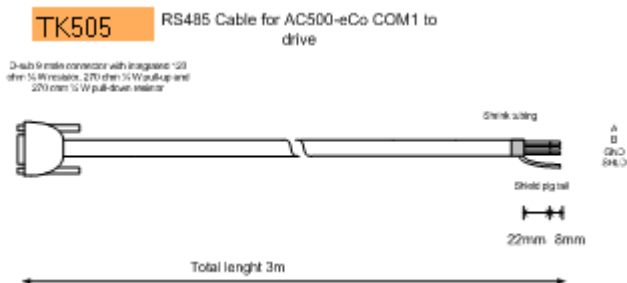
- Installation where total cable length is more than 50 m
  - Use CPU AC500 rather than AC500-eCo or AC500-eCo with TK506 RS 485 isolator for COM1.
- Installation where there is a risk that ground connections in different nodes could vary in voltage potential (typically in ship installations or with cable lengths more than 50 m).
  - Use CPU AC500 and do not connect ground cable, only shield → 2-wire solution, or use AC500-eCo with TK506 RS 485 isolator for COM1.
- When AC500-eCo is used
  - Use of 3-wire connection is recommended because the COM interface is not opto-isolated, or use AC500-eCo with TK506 RS 485 isolator for COM1.
  - Use of cable length more than 50m should be realized with repeaters, or use AC500-eCo with TK506 RS 485 isolator for COM1.
- Cable type
  - Use only cables with twisted pair wires and braided shields. Foil shields are not robust enough, cannot be contacted well and have poor HF (High Frequency) properties.
  - Ground the cable shield directly with a clip when entering the switch-gear cabinet. Do not cut the shield until the cable reaches the connected module.
  - The connection between the PE bar and the shield bar must have low impedance.
  - Use ABB cable TK505 when suitable for easy installation.



Bus line	
Construction	2 cores, twisted, <b>with</b> common <b>shield</b>
Conductor cross section	>0.22 mm <sup>2</sup> (24 AWG)
-recommendation	0.5 mm <sup>2</sup> corresponds to Ø0.8 mm
Twisting rate	<10/meter (symmetrically twisted)
Core insulation	Polyethylene (PE)
Resistance per core	<100 Ω/km
Characteristic impedance	Appr. 120 Ω (100...150 Ω)
Capacitance between the cores	<55 nF/km (if higher, then max bus length must be reduced)
Terminating resistors	120 Ω, ¼ W at both line ends
Remarks	Cables with PVC core insulation and a core diameter of 0.8 mm can be used up to a length of appr. 250 m. In this case, the bus terminating resistor is appr. 100 Ω.

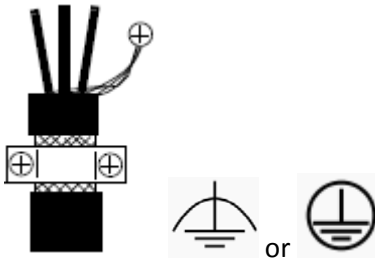
## ABB cable TK505

For an easy Modbus installation in the CPU end, the pre-made ABB cable TK505 can be used. It includes a 9 pole D-sub contact with built-in end termination (120 Ω) and pull-up/pull-down resistors (2\*270 Ω). This contact is suitable for AC500-eCo COM1. It can also be used for AC500 COM2 (even though another voltage level is used), but only if the GND wire is not connected to the Modbus slaves (2-wire solution, preferred for AC500 COM ports).



## Installation examples

Best connection of shield is to connect directly to the mounting plate of the cabinet as near as possible to the AC500. The mounting plate has to be connected to common earth in the cabinet at one point.

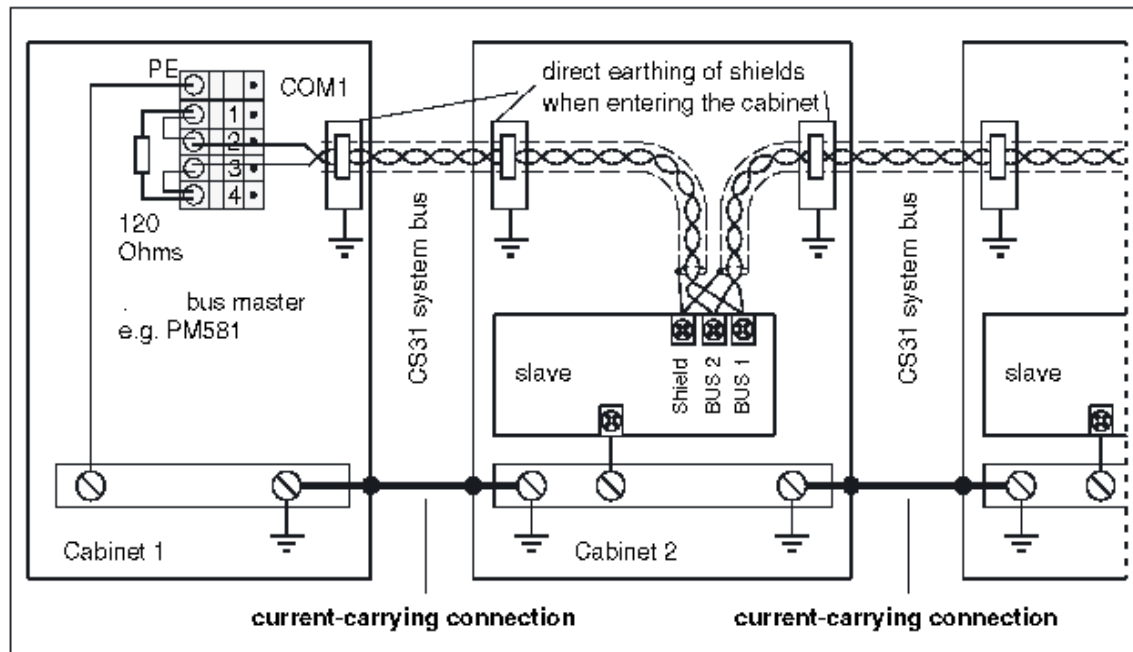


## Earthing

In order to avoid disturbance, the cable shields must be earthed directly.

### Case a:

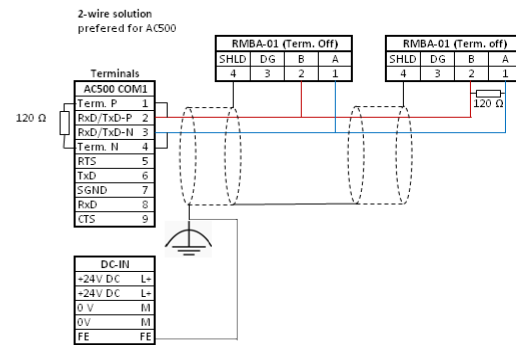
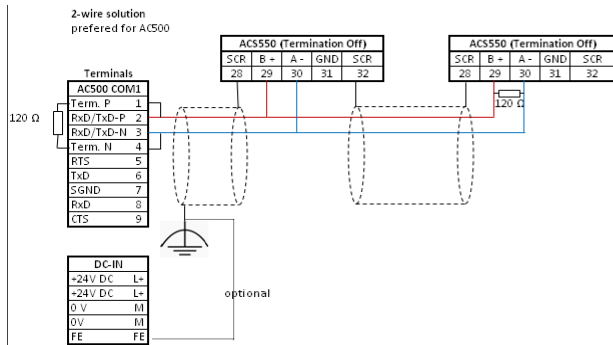
Multiple switch-gear cabinets: If it can be guaranteed that no potential differences can occur between the switch-gear cabinets by means of current-carrying metal connections (earthing bars, steel constructions etc.), the direct earthing is chosen.



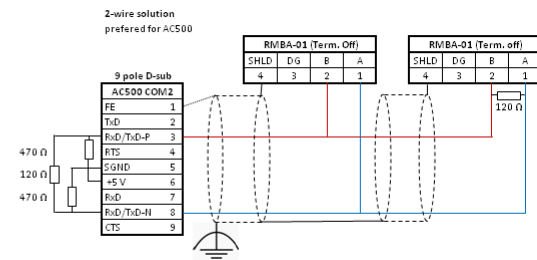
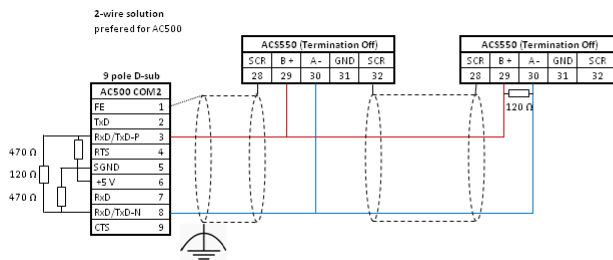
# Quickstart Guide

## ABB PLC and drives integration using Modbus RTU

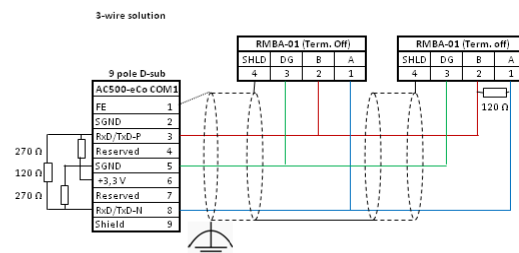
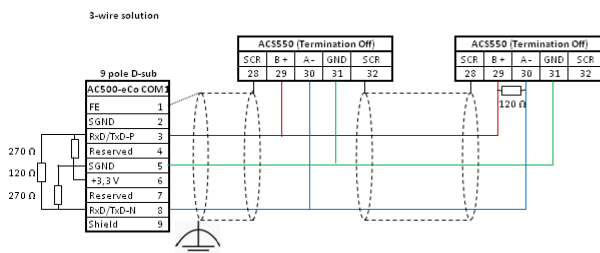
### AC500 COM1 – Master at the bus line end



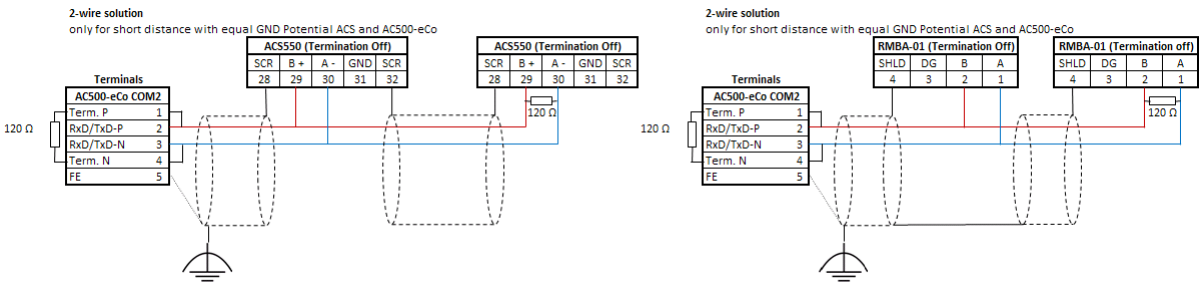
### AC500 COM2 – Master at the bus line end



### AC500-eCo COM1 – Master at the bus line end



AC500-eCo COM2 – Master at the bus line end



## Drive configuration

Drives without embedded Modbus RTU need to be equipped with Modbus RTU adapter according to actual drive type.

The following drive configuration steps will adapt the drive to Modbus RTU control based on ABB-specific drives library PS553-DRIVES.

Drive parameters can be set from the drive's control panel or from drive-specific pc tool (*DriveWindow Light* for ACS355 and ACS550, *DriveStudio* for ACS850 and ACSM1, *Drive Composer* for ACS880 and *DriveWindow* for ACS800).

**Note!** All parameter settings are based on drive default settings. If the drive has been parameterized previously, return to default settings before continuing. It can be done by:

- Changing macro (and then changing back again) in parameter **99.02** for ACS310, ACS355 and ACS550.
- Setting parameter **99.03** *APPLIC RESTORE* to **YES** in ACS800.
- Setting parameter **16.04** *Param restore* to **Restore defs** in ACS850 and ACSM1.
- Setting parameter **96.06** *Parameter restore* to **Restore defaults** in ACS880.

## Starting up ACS310 drives with embedded Modbus RTU

### *Minimum required parameter settings (based on factory default settings)*

Parameter	Description	Setting	Comment
98.02	COMM PROT SEL	STD MODBUS	Activates fieldbus communication
53.02	EFB STATION ID	[Address]	Modbus RTU node address of the drive
53.03	EFB BAUD RATE	19.2 kbit/s (example)	Transfer rate of the link. Same baud rate must be defined in the Modbus RTU master.
53.04	EFB PARITY	8 NONE 1 (example)	Parity and stop bits. Same parity and stop bits must be defined in the Modbus RTU master.
53.05	PROFILE	ABB DRV FULL	Communication profile "ABB Drives"
<b>Drive power down and power up for the new fieldbus settings to take effect!</b>			
10.01	EXT 1 COMMANDS	COMM	Fieldbus interface as source for start and stop
11.02	EXT1/EXT2 SEL	COMM	Fieldbus interface as source to switch to EXT2
11.03	REF1 SELECT	COMM	Fieldbus interface as source for speed reference
16.04	FAULT RESET SEL	COMM	Fieldbus interface as source for fault reset

### *Optional parameter settings*

30.18	COMM FAULT FUNC	[optional]	How the drive reacts to a fieldbus communication break.
30.19	COMM FAULT TIME	[optional]	Time between communication break detection and the selected action.
(11.05)	REF1 MAX	[Scale max]	Max speed/frequency scaling value (used in function block/visualization input "SPEED_REF_MAX"). Must be less or equal to drive parameter max speed/frequency.

## Starting up ACS355 drives with FMBA-01 Modbus RTU adapter

### Minimum required parameter settings (based on factory default settings)

Parameter	Description	Setting	Applic Macro "AC500 MODBUS" *)	Comment
98.02	COMM PROT SEL	STD MODBUS	STD MODBUS	Activates fieldbus module
53.02	EFB STATION ID	[Address]	2	Modbus RTU node address of the drive
53.03	EFB BAUD RATE	19.2 kbit/s (example)	19.2 kbit/s	Transfer rate of the link. Same baud rate must be defined in the Modbus RTU master.
53.04	EFB PARITY	8 NONE 1 (example)	8 NONE 1	Parity and stop bits. Same parity and stop bits must be defined in the Modbus RTU master.
53.05	PROFILE	ABB DRV FULL	ABB DRV FULL	Communication profile "ABB Drives"
53.10	Actual Speed	101	101	Actual speed feedback
53.11	Actual Value2	105	303	Actual value2 (e.g torque 105) (303 = Status Word 1)
<b>Drive power down and power up for the new fieldbus settings to take effect!</b>				
10.01	EXT 1 COMMANDS	COMM	COMM	Fieldbus interface as source for start and stop
11.02	EXT1/EXT2 SEL	COMM	COMM	Fieldbus interface as source to switch to EXT2
11.03	REF1 SELECT	COMM	COMM	Fieldbus interface as source for speed reference
16.04	FAULT RESET SEL	COMM	COMM	Fieldbus interface as source for fault reset

### Optional parameter settings

30.18	COMM FAULT FUNC	[optional]	FAULT	How the drive reacts to a fieldbus communication break.
30.19	COMM FAULT TIME	[optional]	Default:.. 3s	Time between communication break detection and the selected action.
(11.05)	REF1 MAX	[Scale max]		Max speed/frequency scaling value (used in function block/visualization input "SPEED_REF_MAX"). Must be less or equal to drive parameter max speed/frequency.

\*) in Par. 99.02 the Application Macro can be set to "AC500 Modbus (10)" then the parameters are automatically set as shown.

## Starting up ACS550 with embedded Modbus RTU

### *Minimum required parameter settings (based on factory default settings)*

Parameter	Description	Setting	Comment
98.02	COMM PROT SEL	EXT FBA	Activates fieldbus communication
53.02	EFB STATION ID	[Address]	Modbus RTU node address of the drive
53.03	EFB BAUD RATE	19.2 kbit/s (example)	Transfer rate of the link. Same baud rate must be defined in the Modbus RTU master.
53.04	EFB PARITY	8 NONE 1 (example)	Parity and stop bits. Same parity and stop bits must be defined in the Modbus RTU master.
53.05	PROFILE	ABB DRV FULL	Communication profile "ABB Drives"
<b>Drive power down and power up for the new fieldbus settings to take effect!</b>			
10.01	EXT 1 COMMANDS	COMM	Fieldbus interface as source for start and stop
11.02	EXT1/EXT2 SEL	COMM	Fieldbus interface as source to switch to EXT2
11.03	REF1 SELECT	COMM	Fieldbus interface as source for speed reference
16.04	FAULT RESET SEL	COMM	Fieldbus interface as source for fault reset

### *Optional parameter settings*

30.18	COMM FAULT FUNC	[optional]	How the drive reacts to a fieldbus communication break.
30.19	COMM FAULT TIME	[optional]	Time between communication break detection and the selected action.
(11.05)	REF1 MAX	[Scale max]	Max speed/frequency scaling value (used in function block/visualization input "SPEED_REF_MAX"). Must be less or equal to drive parameter max speed/frequency.



## Starting up ACS800 drives with RMBA-01 Modbus RTU adapter

### *Minimum required parameter settings (based on factory default settings)*

Parameter	Description	Setting	Comment
98.02	COMM. MODULE LINK	STD MODBUS	Activates fieldbus module
98.07	COMM PROFILE	ABB DRIVES	Communication profile “ABB Drives”
52.01	STATION NUMBER	[Address]	Modbus RTU node address of the drive
52.02	BAUDRATE	19.2 kbit/s (example)	Transfer rate of the link. Same baud rate must be defined in the Modbus RTU master.
52.03	PARITY	NONE1STO PBIT (example)	Parity and stop bits. Same parity and stop bits must be defined in the Modbus RTU master.
<b>Drive power down and power up for the new fieldbus settings to take effect!</b>			
10.01	EXT 1 STRT/STP/DIR	COMM.CW	Fieldbus interface as source for start and stop
11.02	EXT1/EXT2 SELECT	COMM.CW	Fieldbus interface as source to switch to EXT2 control location
11.03	EXT REF1 SELECT	COMM.REF	Fieldbus interface as source for speed reference
16.04	FAULT RESET SEL	COMM.CW	Fieldbus interface as source for fault reset
92.02	MAIN DS ACT1	102	Actual speed value taken from 01.02
92.02	MAIN DS ACT2	e.g. 105	Actual value2. E.g torque taken from 01.05

### *Optional parameter settings*

30.18	COMM FLT FUNC	[optional]	How the drive reacts to a fieldbus communication break.
30.19	MAIN REF DS T-OUT	[optional]	Time between communication break detection and the selected action.
(11.05)	EXT REF1 MAXIMUM	[Scale max]	Max speed/frequency scaling value (used in function block/visualization input “SPEED_REF_MAX”). Must be less or equal to drive parameter max speed/frequency.

## Starting up ACS850, ACQ810 drives with embedded Modbus RTU

### *Minimum required parameter settings (based on factory default settings)*

Parameter	Description	Setting	Comment
50.15	Fb cw used	P.02.36	Selects the address of the fieldbus Control Word in use.
58.01	Protocol ena sel	Modbus RTU	Initializes embedded fieldbus communication. Drive-to-drive link operation (parameter group 57) is automatically disabled.
58.03	Node address	[Address]	Modbus RTU node address of the drive
58.04	Baud rate	19.2 kbit/s (example)	Transfer rate of the link. Same baud rate must be defined in the Modbus RTU master.
58.05	Parity	8 none 1 (example)	Parity and stop bits. Same parity and stop bits must be defined in the Modbus RTU master.
58.06	Control profile	ABB Classic	Communication profile "ABB Drives"
58.10	Refresh settings	[optional]	Refreshes the settings of parameters 58.01...58.09.
10.01	Ext1 start func	FB	Fieldbus interface as source for start and stop
12.01	EXT1/EXT2 SEL	P.02.36 bit 15	Fieldbus interface as source to switch to EXT2 control location
21.01	Speed ref1 sel	EFB ref1	Fieldbus interface as source for speed reference
10.10	Fault reset sel	P.02.36 bit 8	Fieldbus interface as source for fault reset

### *Optional parameter settings*

58.07	Comm loss t out	[optional]	Defines the timeout limit for the EFB communication monitoring.
58.08	Comm loss mode	None or Any Message	Enables/disables EFB communication loss monitoring and defines the means for resetting the counter of the communication loss delay.
58.09	Comm loss action	[optional]	Defines the drive operation after the EFB communication loss monitoring awakes.
58.10	Refresh settings	[optional]	Refreshes the settings of parameters 58.01...58.09.
(19.01)	Speed scaling	[Scale max]	Max speed/frequency scaling value (used in function block/visualization input "SPEED_REF_MAX"). Must be less or equal to drive parameter max speed/frequency.

## Starting up ACS880 drives with FSCA-01 Modbus RTU adapter

### *Minimum required parameter settings (based on factory default settings)*

Parameter	Description	Setting	Comment
50.01	FBA A enable	Enable	Activates fieldbus module
51.02	PROTOCOL/PROFILE	ABB Drives Classic or Enhanced	Communication profile "ABB Drives"
51.03	STATION ID	[Address]	Modbus RTU node address of the drive
51.04	BAUD RATE	19.2 kbit/s (example)	Transfer rate of the link. Same baud rate must be defined in the Modbus RTU master.
51.05	EFB PARITY	8 NONE 1 (example)	Parity and stop bits. Same parity and stop bits must be defined in the Modbus RTU master.
52.02	FBA data in2	Act1 16bit (5)	Actual speed as Data Word 2 from the drive
53.02	FBA data out2	Ref1 16bit (2)	Speed reference as Data Word 2 to the drive
<b>51.27</b>	FBA par refresh	Configure	Updates fieldbus settings (groups 50 to 57)
19.11	Ext1/Ext2 selection	Mcw Bit11 (06.01)	Fieldbus interface as source to switch to EXT2 control location
20.01	Ext 1 commands	Fieldbus A	Fieldbus interface as source for start and stop
20.02	Ext1 start trigger	Level	To directly restart after Coast Stop (Off3)
22.11	Speed ref1 selection	FB A ref1	Fieldbus interface as source for speed reference
31.11	Fault reset selection	P.06.01 bit 7	Fieldbus interface as source for fault reset

### *Optional parameter settings*

50.02	FBA A comm loss func	[optional]	Defines the drive operation after the communication loss monitoring awakes.
51.06	MODBUS TIMEOUT	[optional]	Time between communication break detection and the selected action. 1 = 100 ms.
51.07	TIMEOUT MODE	CONTROL WRITE	How the drive reacts to a fieldbus communication break.
<b>51.27</b>	FBA PAR REFRESH	REFRESH	Updates fieldbus settings (groups 50 to 53)
(46.01)	Speed scaling	[Scale max]	Max speed/frequency scaling value (used in function block/visualization input "SPEED_REF_MAX"). Must be less or equal to drive parameter max speed/frequency.

## Starting up ACSM1 drives with FSFA-01 Modbus RTU adapter

### *Minimum required parameter settings (based on factory default settings)*

Parameter	Description	Setting	Comment
50.01	FBA ENABLE	Enable	Activates fieldbus module
51.02	PROTOCOL/PROFILE	ABB Drives Classic	Communication profile "ABB Drives"
51.03	STATION ID	[Address]	Modbus RTU node address of the drive
51.04	BAUD RATE	19.2 kbit/s (example)	Transfer rate of the link. Same baud rate must be defined in the Modbus RTU master.
51.05	EFB PARITY	8 NONE 1 (example)	Parity and stop bits. Same parity and stop bits must be defined in the Modbus RTU master.
52.01	FBA DATA IN1	4	Status word as Data Word 1 from the drive
52.02	FBA DATA IN2	5	Actual speed as Data Word 2 from the drive
53.01	FBA DATA OUT1	1	Control word as Data Word 1 to the drive
53.02	FBA DATA OUT2	2	Speed reference as Data Word 2 to the drive
<b>51.27</b>	FBA PAR REFRESH	REFRESH	Updates fieldbus settings (groups 50 to 53)
10.01	EXT1 START FUNC	FBA	Fieldbus interface as source for start and stop
12.01	EXT1/EXT2 SEL	P.02.12 bit15	Fieldbus interface as source to switch to EXT2 control location
24.01	SPEED REF1 SEL	FBA REF1	Fieldbus interface as source for speed reference
10.08	FAULT RESET SEL	P.02.12 bit 8	Fieldbus interface as source for fault reset

### *Optional parameter settings*

51.06	MODBUS TIMEOUT	[optional]	Time between communication break detection and the selected action. 1 = 100 ms.
51.07	TIMEOUT MODE	CONTROL WRITE	How the drive reacts to a fieldbus communication break.
50.02	COMM LOSS FUNC	[optional]	Defines the drive operation after the communication loss monitoring awakes.
<b>51.27</b>	FBA PAR REFRESH	REFRESH	Updates fieldbus settings (groups 50 to 53)
(25.02)	SPEED SCALING	[Scale max]	Max speed/frequency scaling value (used in function block/visualization input "SPEED_REF_MAX"). Must be less or equal to drive parameter max speed/frequency.

## Install drives library

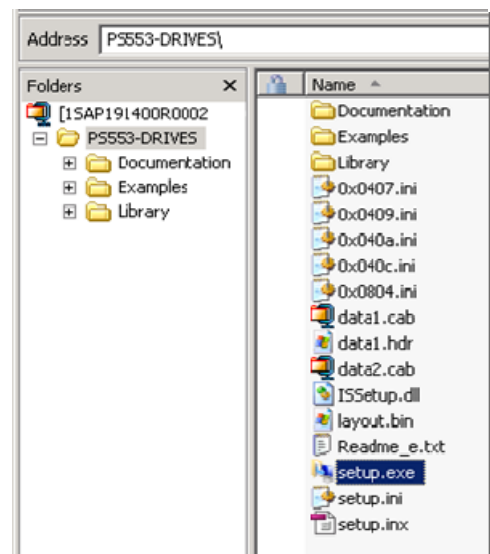
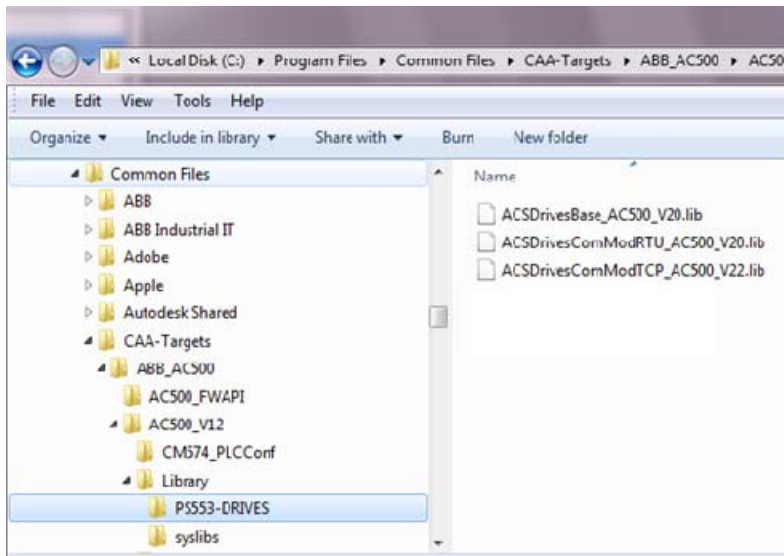
**Note!** In Control Builder Plus V2.3 and later, the ACSDrivesBase library is already installed. You can find it in the subfolder “PS553-DRIVES” of the default library folder (see left picture below)

**Note!** The version of the library can differ, nevertheless the name will stay the same! Creation date and main changes can be seen in the library manager of CODESYS only.

For older Versions of Control Builder Plus you can download the installation package in the following way from:

- [www.abb.com/PLC](http://www.abb.com/PLC):
- On right side menu under heading "Your preferences": Select "English" as language ... (country doesn't matter).
- On right side menu under "More Info Links" : Click on "PS501 Updates".
- Select "PS501-UPDA: PS553-DRIVES..." to download \*.zip file.

Install the drives library “PS553-DRIVES” with ready-made function blocks and visualization objects by unzipping to a temporary directory and clicking on the file “setup.exe” (see right picture below).

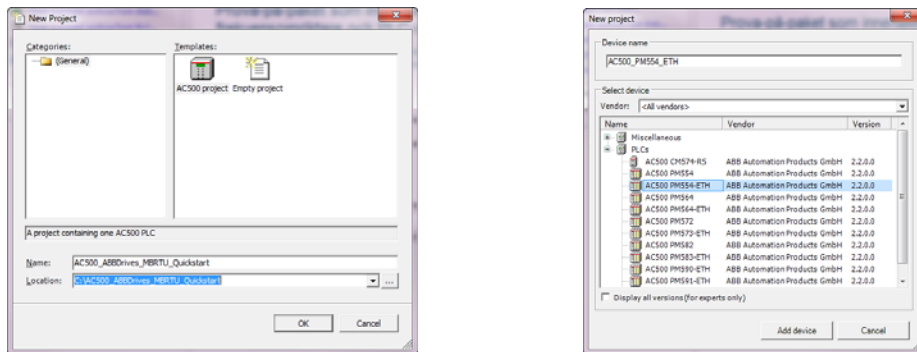


## Control Builder Plus for PLC and Drives

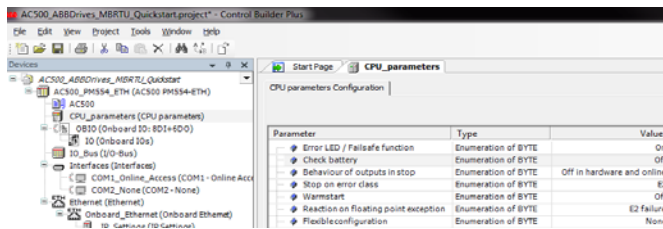
Start the “Control Builder Plus” PC tool for PLC configuration and go through the following steps.

### Hardware configuration

File → New Project → Choose “AC500 project” and name the project. Choose PLC device according to your equipment and then “Add device”.



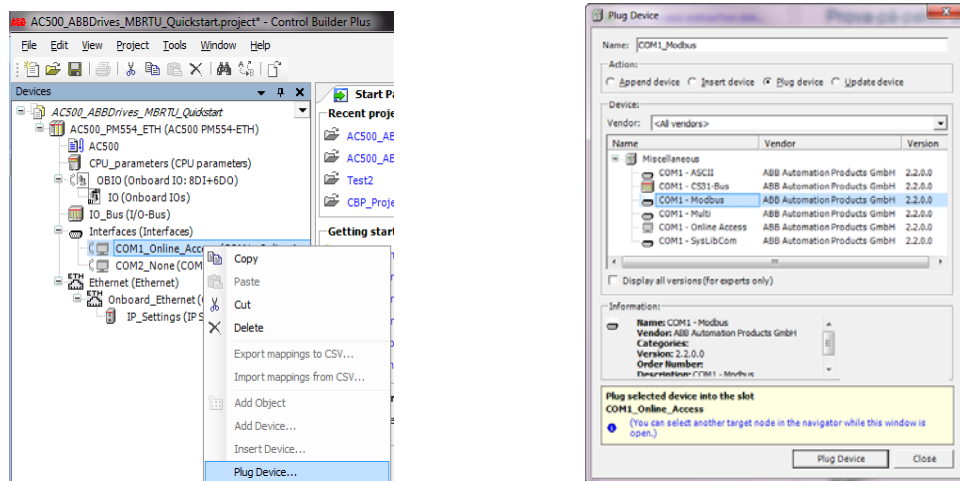
Double-click “CPU\_parameters” and configure according to your application. Set for instance “Check battery” = “Off” if your CPU does not have a battery installed.



Right-click actual COM interface (COM1 is used in the example) and click “Plug Device”. Select “COMX – Modbus” and click “Plug Device”.

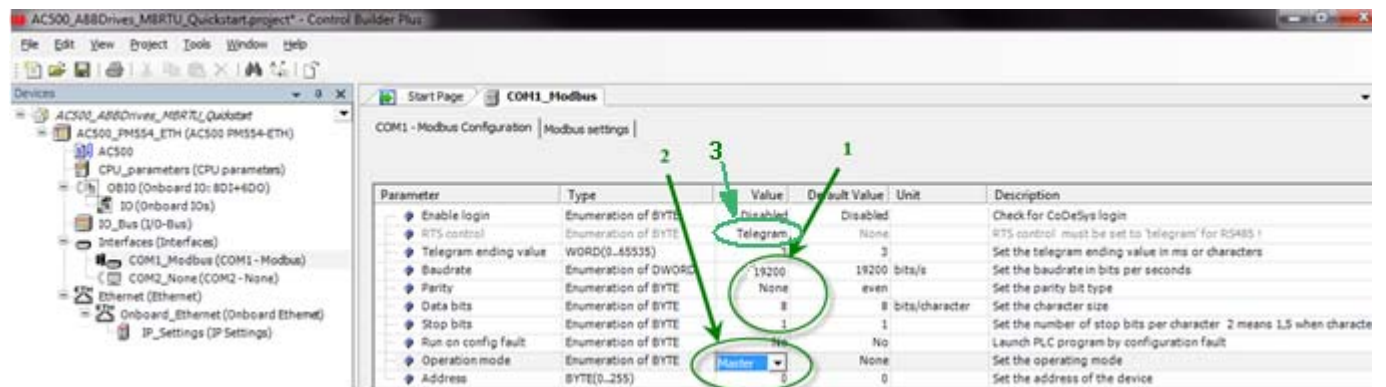
## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU



Double-click “COMX\_Modbus” to configure the Modbus settings. Do at least the following settings:

- Set “Baudrate”, “Parity”, “Data bits” and “Stop bits” (1) according to settings in actual drive.
  - Set “Operation mode” to “Master” and “Address” to “0” (2).
  - Set “RTS control” to “Telegram” (3)
- (=RS485 - in AC500-eCo this is already set)

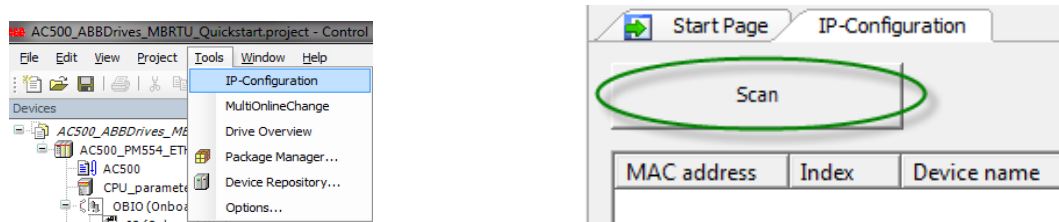


## IP configuration of CPU (for Ethernet connection between PC and PLC)

If the CPU has an Ethernet port and the CPU IP address has not already been configured, follow the steps below.

De-activate the firewall of the PC, or allow the connection if question comes up. Connect an Ethernet cable between PC and PLC (or via a switch).

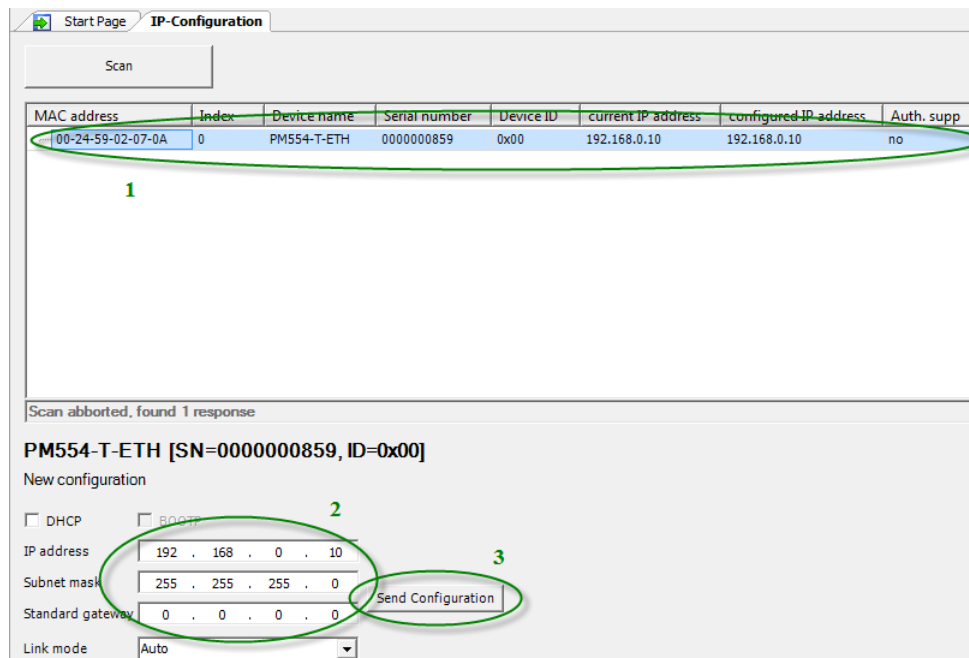
Open the IP configuration tool from the “Tools” menu and press the “Scan” button.



Select actual PLC from the scanned objects (1), make your network settings (2) and press the “Send Configuration” button. This needs to be done the first time you connect to the PLC even if the IP address is already correct.

**Note!** The PLC needs to be in “STOP” mode for IP configuration.

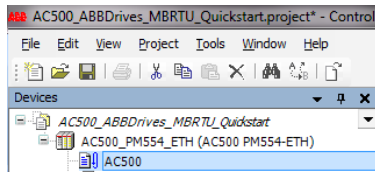
After the new configuration was sent the PLC will automatically reboot.





## CODESYS

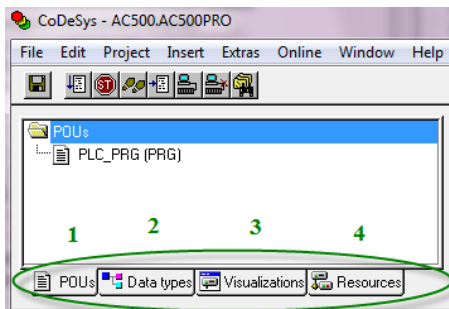
Double-click “AC500” from the Device tree in the Control Builder Plus project to open CODESYS.



## CODESYS tabs

In CODESYS there are four different tabs for programming, configuration, etc.

1. POU (Program Organization Units): This tab contains your Functions, Function blocks and Programs.
2. Data types: In this tab, along with the standard data types you can define your own user-specific data types.
3. Visualizations: In this tab you can create your own or use predefined visualizations to display your project variables graphically. In Online mode, these can then change their form, color, text, position, output, etc. in response to specified variable values.
4. Resources: In this tab you can configure and organize your project. It includes Global variables, Library manager, etc.

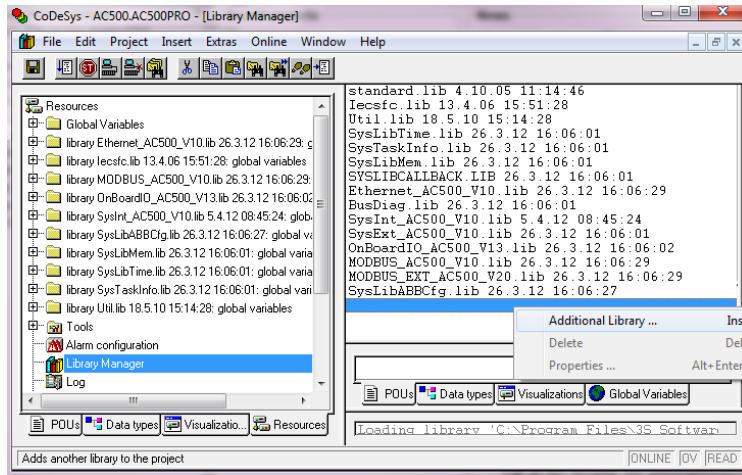


## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

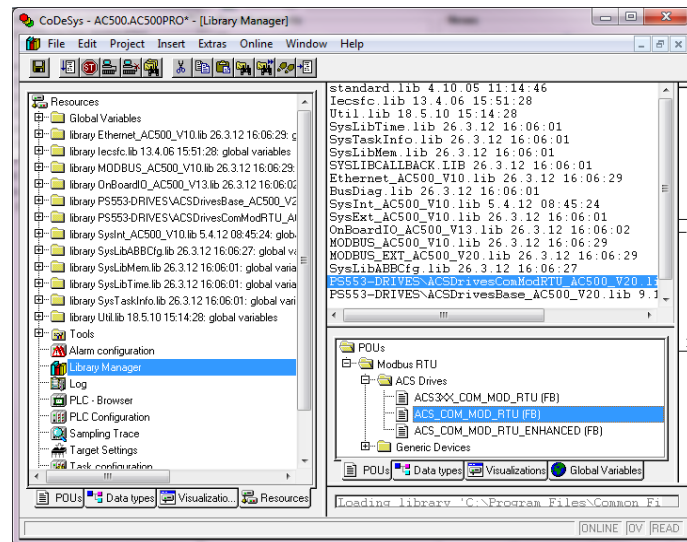
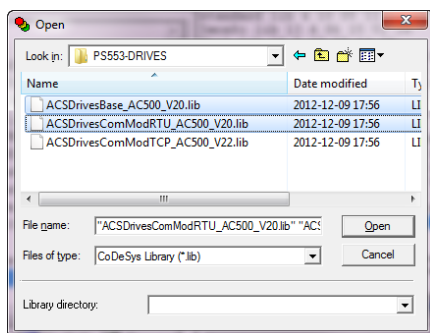
#### Library Manager

Open the Library Manager by double-clicking “Library Manager” from the “Resources” tab. Right-click in the library field and choose “Additional Library”.



Select the “ACSDrivesBase\_AC500\_V20.lib” and “ACSDrivesComModRTU\_AC500\_V20.lib” library files from the PS553-DRIVES catalog (under the standard CODESYS library catalog) and click “Open”. The libraries are added to the project.

**Note!** When copying projects the used libraries must be found on the same library path as on the original PC. Otherwise a warning will show that a library cannot be found. If so, this red marked library has to be deleted and added newly from the now different path to the project.



---

## Quickstart Guide

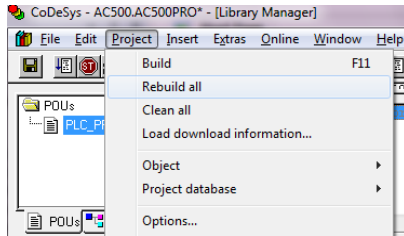
### ABB PLC and drives integration using Modbus RTU

---

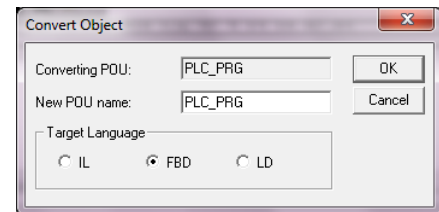
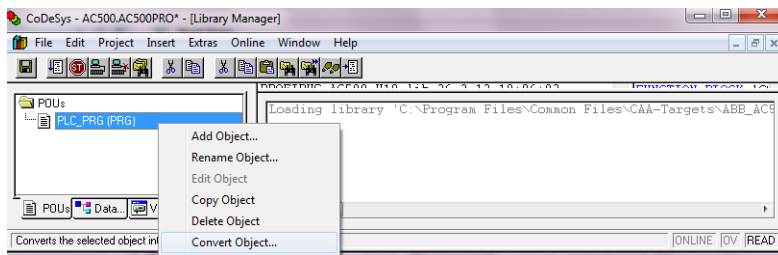
#### PLC logic

The following program handles a Speed control drive application.

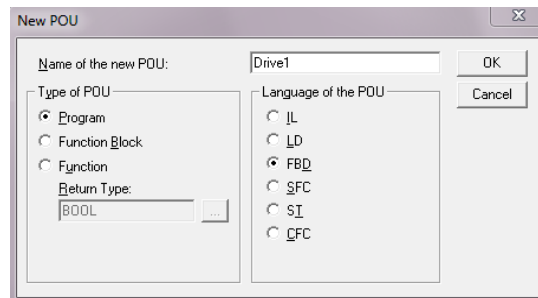
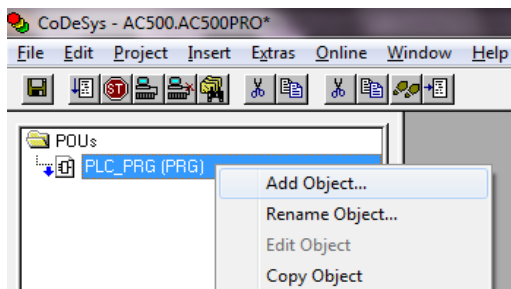
Compile your project, choose “Rebuild all” from the “Project” menu.



Right-click “PLC\_PRG” in the “POUs” tab and choose “Convert Object”. Choose Target Language “FBD” and click “OK”.



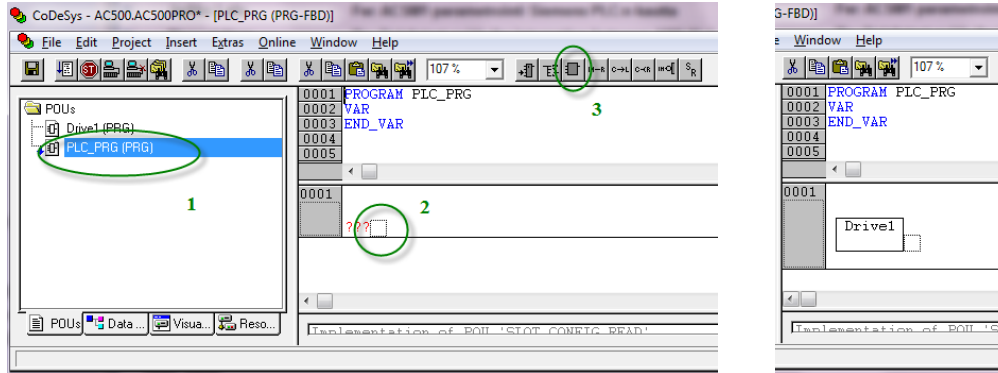
Right-click in the POU's field and choose “Add Object”. Set Type of POU to “Program” and Language of the POU to “FBD”, give the new Program a suitable name and click “OK”.



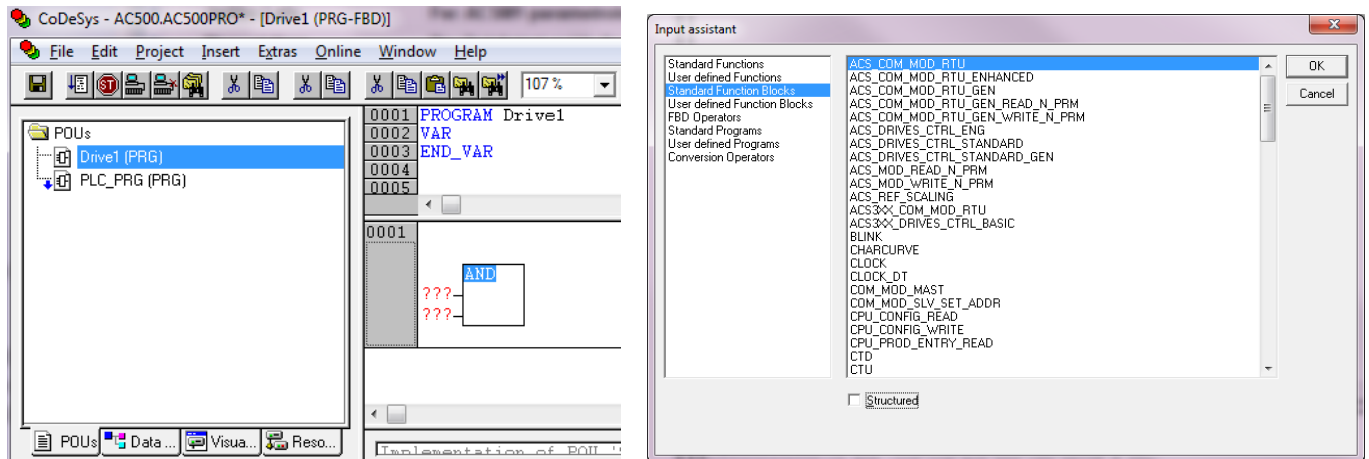
## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

Double-click “PLC\_PRG” (1) to open the main program. Select the dotted box (2) in Network 0001 and insert a box (3). Write the name of your new Program (Drive1 (PRG)) in the example) to call for it from the main program.



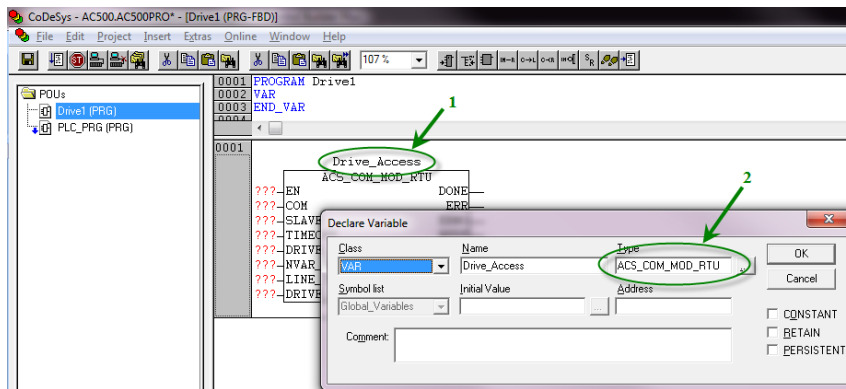
Double-click your new program and add a box as described above. This time, press F2 while the block title is selected and choose “ACS\_COM\_MOD\_RTU” from “Standard Function Blocks”, then click “OK” (Tip: untick the “Structured” box in the Input assistant).



Give the instance of the drive access block a suitable name (1) and declare the variable of type “ACS\_COM\_MOD\_RTU” (2) (automatic suggestion).

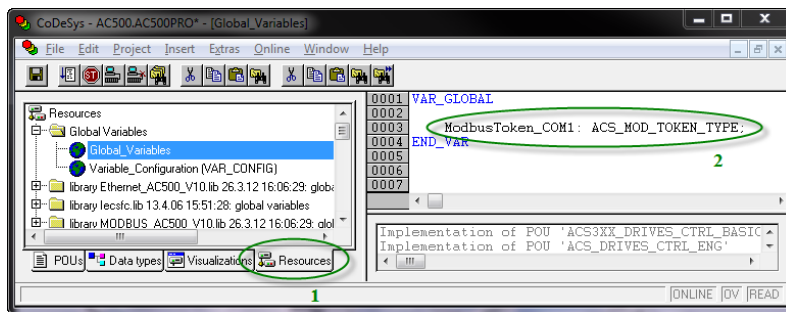
## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU



Double-click “Global\_Variables” in the “Resources” tab (1) and create a global variable for the Modbus token handling of type “ACS\_MOD\_TOKEN\_TYPE” (2). This variable will be used for connecting the drive to a certain Modbus line.

**Note!** This variable should be global (One single variable used for all drives).



Open the program again from the “POUs” tab and connect the function block inputs and outputs according to your needs. In the example below, the block will always be enabled, COM port 1 of the PLC is used, the drive has Modbus node number 1, default timeout 1000 ms is used, drive type is ACS355, no extra parameters besides Status word and Actual speed/torque are read from the drive.

The variables declared between “VAR” and “END\_VAR” in the picture below are local variables and can only be accessed from the actual program (PRG) or from the Visualization objects (see chapter “Visualizations”).

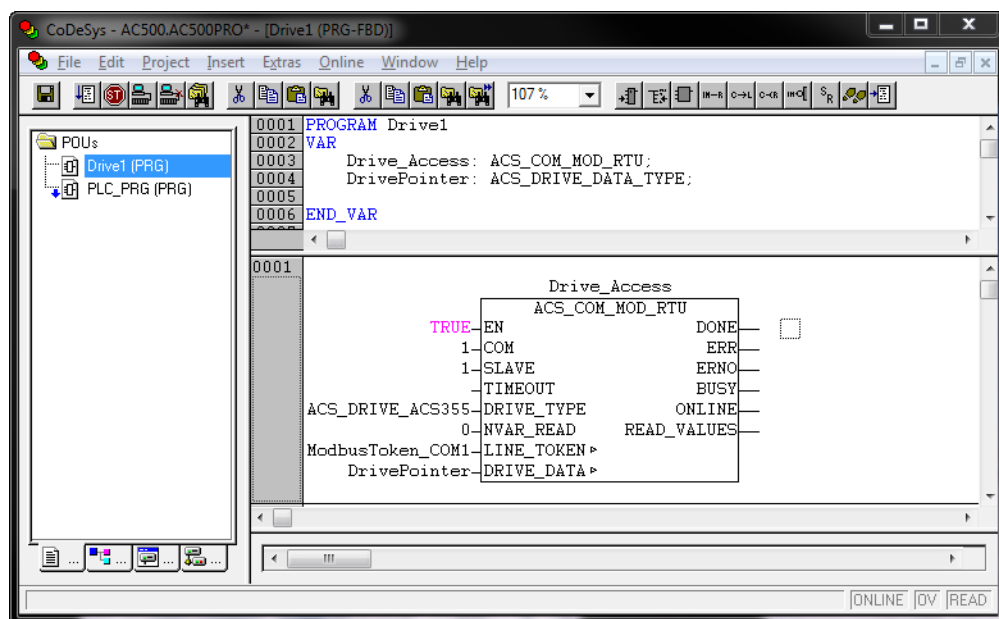
**Tip!** Click inside the name of the function block and press F1 to open the description of the block in the online help.

---

## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

---

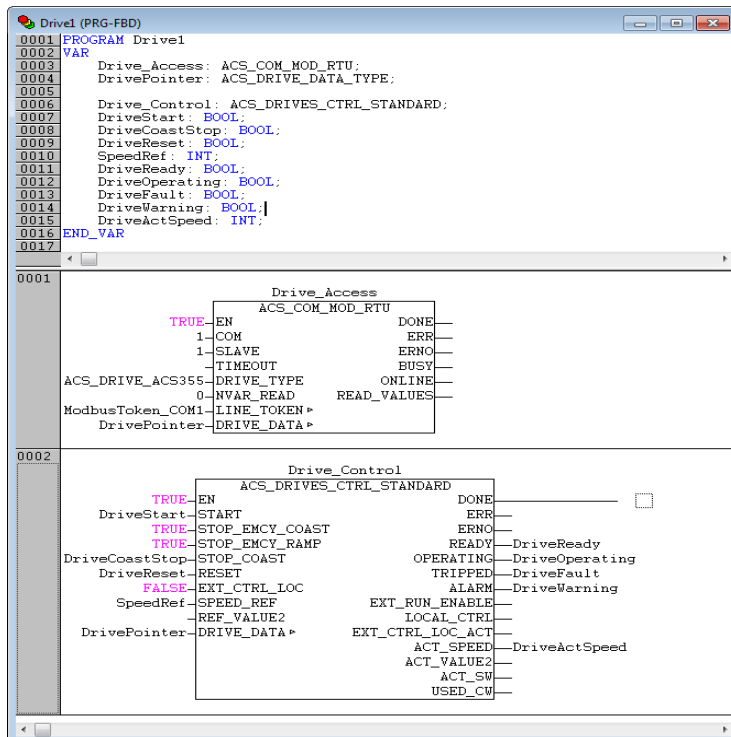


Create a second network (Ctrl +T) in the same program and add the block “ACS\_DRIVES\_CTRL\_STANDARD” in the same way as for “ACS\_COM\_MOD\_RTU”. Connect the function block inputs and outputs according to your needs. In the example below:

- The block will always be enabled
- The drive is started by the variable “DriveStart”
- Emergency stop functions are not used in this example
- Ramp stop by setting “DriveStart” = FALSE
- Coast stop by setting “DriveCoastStop” = TRUE
- Fault reset by setting “DriveResetFault” = TRUE (positive edge).
- Speed reference is set by the variable “SpeedRef” as a value between -20000 and 20000, where 20000 corresponds to a parameter in the drive (see chapter “Drive configuration”).

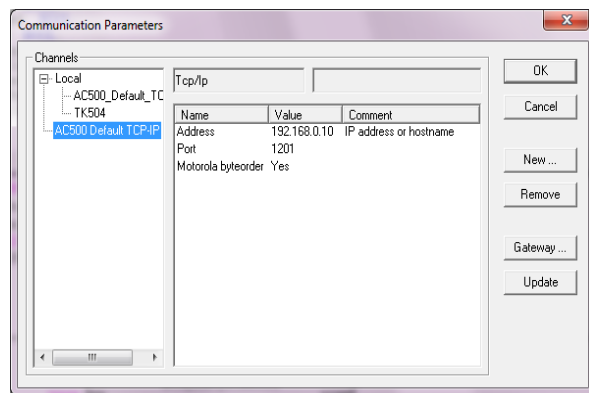
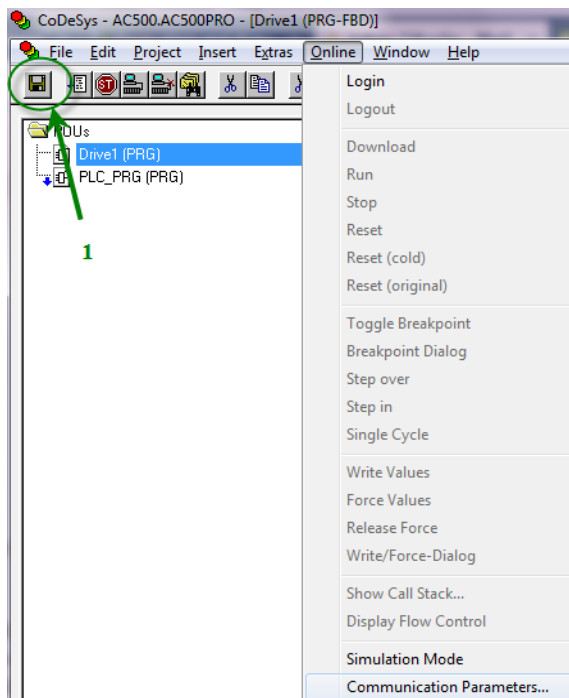
The variables declared between “VAR” and “END\_VAR” in the picture below are local variables inside this program. (To access them from another part e.g. any visualization the programs name has to be used followed by a dot and the variables name. E.g. “Drive1.DriveOperating”

**Note!** The variable connected to “ACS\_DRIVES\_CTRL\_STANDARD” → “DRIVE\_DATA” must be the same as the one connected to “ACS\_COM\_MOD\_RTU” → “DRIVE\_DATA” and must be of type “ACS\_DRIVE\_DATA\_TYPE”.



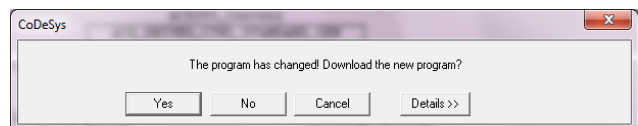
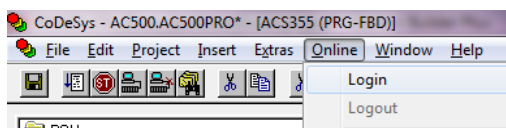
#### Download program to PLC

Save the program and choose “Communication Parameters” from the “Online” menu. Set communication Parameters according to your online connection. In the example below, the “Address” 192.168.0.10 corresponds to the IP address of actual CPU and is also the default IP address of AC500 CPUs. Note that in case of Ethernet connection, the IP address of your PC port or Ethernet adapter should belong to the same subnet as the CPU, 192.168.0.XXX (XXX between 1 and 255).



Choose “Login” from the “Online” menu and in the following pop-up window, click “Yes”. Then choose “Run” from the “Online” menu to start the PLC. Check that the PLC goes to “RUN” mode in the PLC display or led.

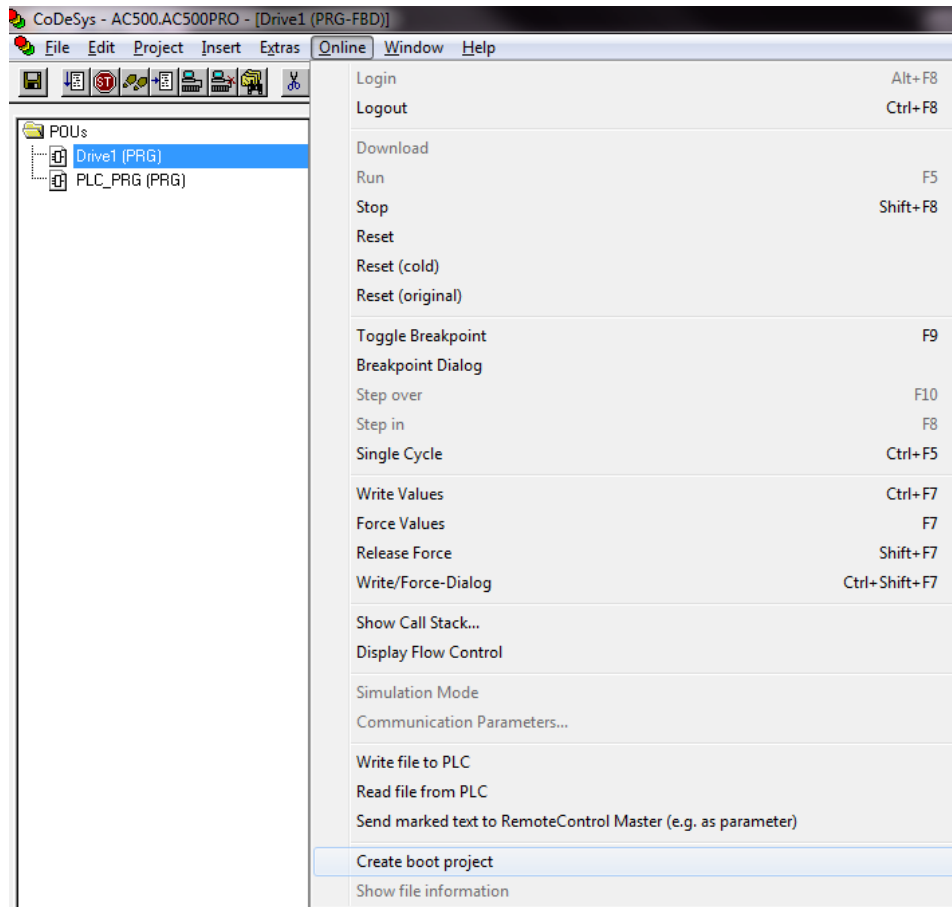
**Note!** If the PLC does not go to “RUN” mode, you might have some PLC errors that you need to reset. You can do that either by pressing the “DIAG” button on the PLC itself (not in the eCo series) followed by the “OK” button until all errors are reset. You can also do it in online mode by using the “diagreset” command from the “PLC Browser” in the CODESYS “Resources” tab.





#### Create boot project

In “online” mode (Login), choose “Create boot project” from the “Online” menu. With this command, the compiled project is stored to the flash in such a way that the PLC will load it automatically when restarted.

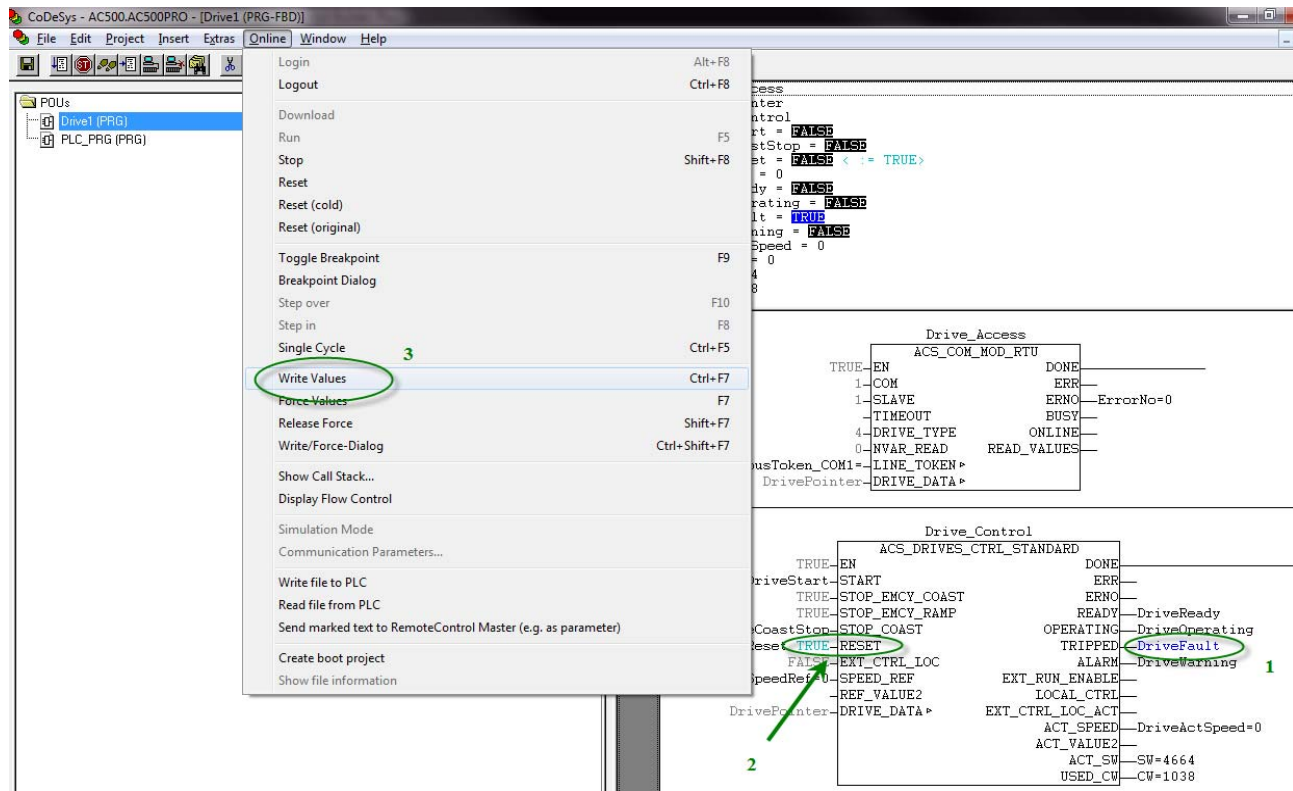


## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

#### Test the program

If the drive has an active fault ("TRIPPED" output is colored blue) (1), then try to reset by setting the "RESET" input = TRUE (2). Double-click the input variable and choose "Write Values" from the "Online" menu (3).

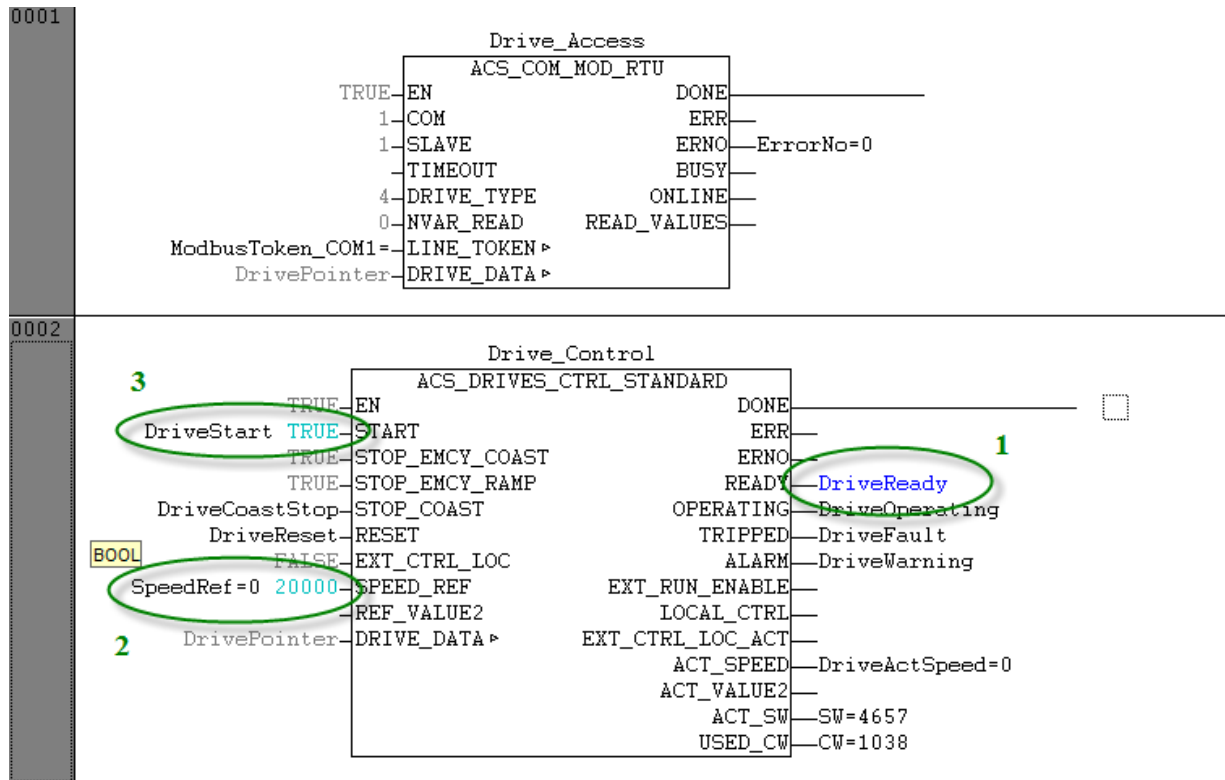


When the "READY" output is TRUE (blue) (1), then set the "SPEED\_REF" input (2) to desired speed value and "START" input = TRUE (3). Write values from the Online menu or Ctrl+F7. Check that the drive starts and follows the given speed reference.

**Note!** Speed reference is set in the block input "SPEED\_REF" as a value between -20000 and 20000, where 20000 corresponds to a parameter in the drive (see chapter "Drive configuration").

## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU



### Read and write more data between PLC and drive (optional)

While the already implemented function block “ACS\_COM\_MOD\_RTU” has built-in functionality to read more data/parameters from pre-determined Modbus register addresses in the drive, the function blocks “ACS\_MOD\_READ\_N\_PRM” and “ACS\_MOD\_WRITE\_N\_PRM” can read and write data/parameters from Modbus register addresses according to user choice.

#### Example - Read drive data with “ACS\_COM\_MOD\_RTU”

Values are read from Modbus register addresses 40007.. 40030. The number of read parameters depends on the function block input “NVAR\_READ”. Configuration is for instance made in parameters 53.12 to 53.17 for ACS310, ACS355 and ACS550, or in parameters 58.35 to 58.58 for ACS850 and ACQ810. The supported number of parameters that can be read in this way depend on actual drive type, for instance 6 parameters can be read from ACS355 or 24 parameters can be read from ACS850.

Additional parameter settings in the drive are necessary, see example below with an ACS355 drive:

- 53.12 = 107 → Modbus register 40007 will consist of ACS355 parameter 1.07 DC voltage.
- 53.13 = 110 → Modbus register 40008 will consist of ACS355 parameter 1.10 Drive temp.
- 53.14 = 120 → Modbus register 40009 will consist of ACS355 parameter 1.20 Analog input 1.

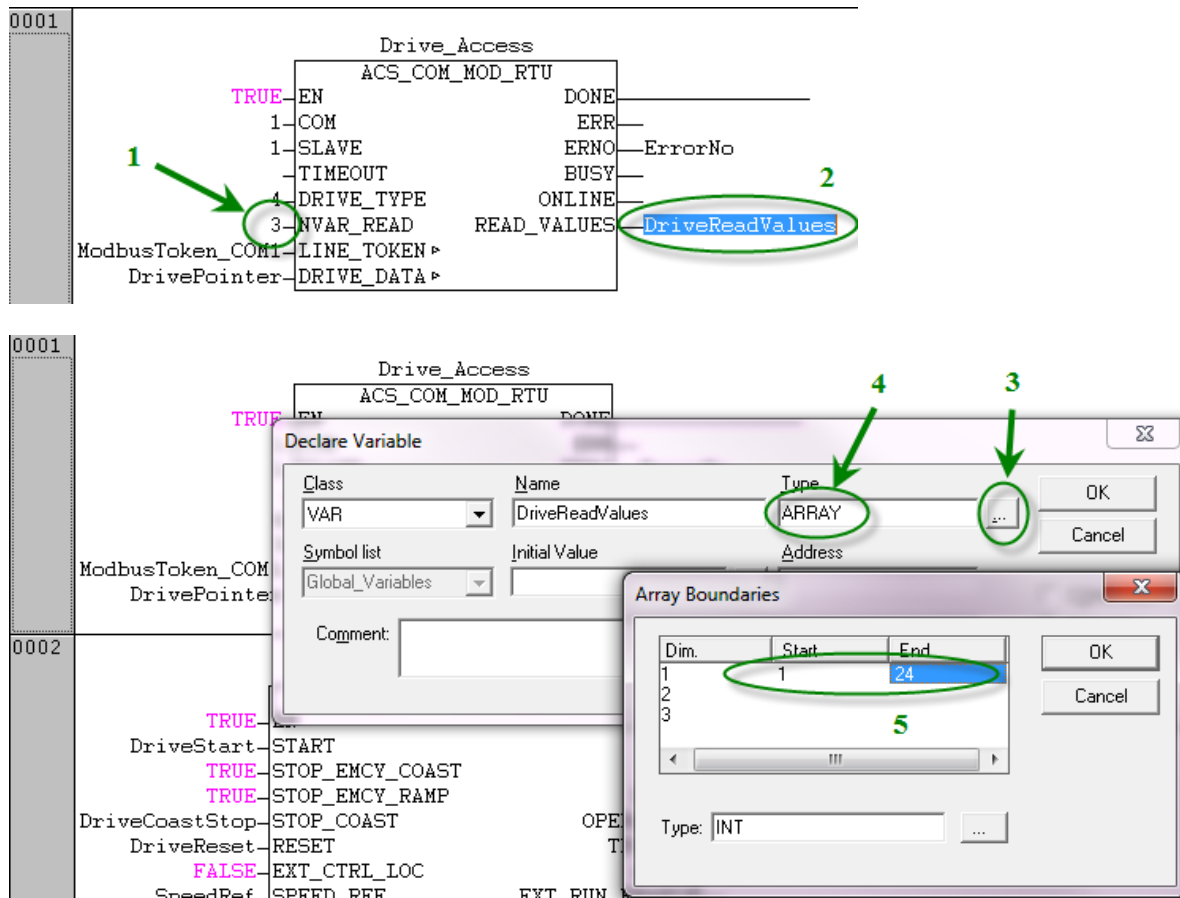
**Note!** After these parameter settings, the drive needs a re-boot (power off/power on) to activate the settings.

53	EFB PROTOCOL			
02	EFB STATION ID	1	1	247
03	EFB BAUD RATE	96	0	0
04	EFB PARITY	8 NONE 1	0	3
05	EFB CTRL PROFILE	ABB DRV FULL	0	2
06	EFB OK MESSAGES	39400	0	65535
07	EFB CRC ERRORS	134	0	65535
10	EFB PAR 10	102	0	65535
11	EFB PAR 11	105	0	65535
12	EFB PAR 12	107	0	65535
13	EFB PAR 13	110	0	65535
14	EFB PAR 14	120	0	65535

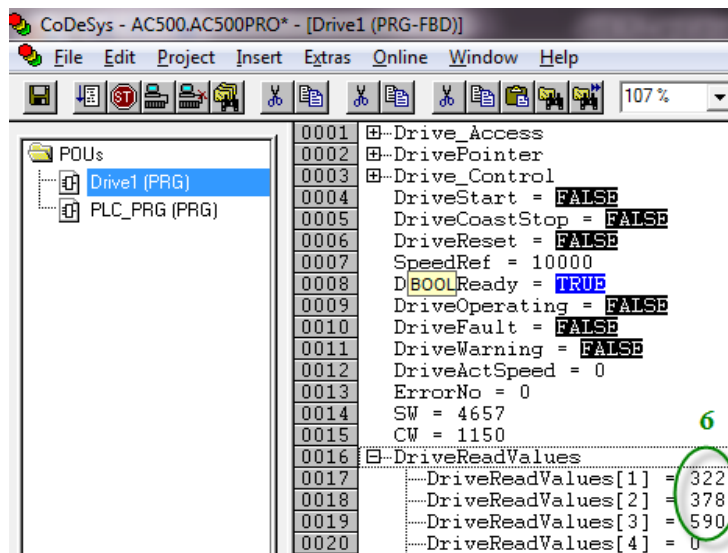
Set “ACS\_COM\_MOD\_RTU” block input “NVAR\_READ” according to number of parameters to be read (1). Create a new variable connected to block output “READ\_VALUES” (2) and press the “...” button (3) to set the variable of type “ARRAY” (4) of INT. Give the array Start index 1 and End index 24 (5), press OK.

## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU



The parameter values from the drive are then written to the new array variable (6).



## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

#### Example - Read drive data with “ACS\_MOD\_READ\_N\_PRM”

Values are read from Modbus register addresses according to user choice. No additional parameter settings in the drive are necessary.

Create a new network in the same program (POU) and add the block “ACS\_MOD\_READ\_N\_PRM”.

The screenshot displays the CoDeSys software interface for an ABB PLC and drives integration project. The main window shows a ladder logic network (Network 0002) with the following components:

- Network 0002:** A network containing the **ACS\_MOD\_READ\_N\_PRM** block. The block is configured with the following parameters:
  - Drive Access:** ACS\_COM\_MOD\_RTU
  - EN:** TRUE
  - COM:** 1
  - SLAVE:** 1
  - TIMEOUT:** 4
  - DRIVE\_TYPE:** 4
  - READ\_VALUES:** 2
  - DriveDataValues:** 2

Two dialog boxes are open over the network:

- Input assistant:** This dialog box is used to select the function block. The **Standard Functions** tab is selected, and the **ACS\_MOD\_READ\_N\_PRM** block is highlighted in the list.
- Declare Variable:** This dialog box is used to declare a variable. The **Class** is set to **VAR**, the **Name** is **Drive\_Mod\_Read1**, and the **Type** is **ACS\_MOD\_READ\_N\_PR**. The **Symbol list** is set to **Global\_Variables**.

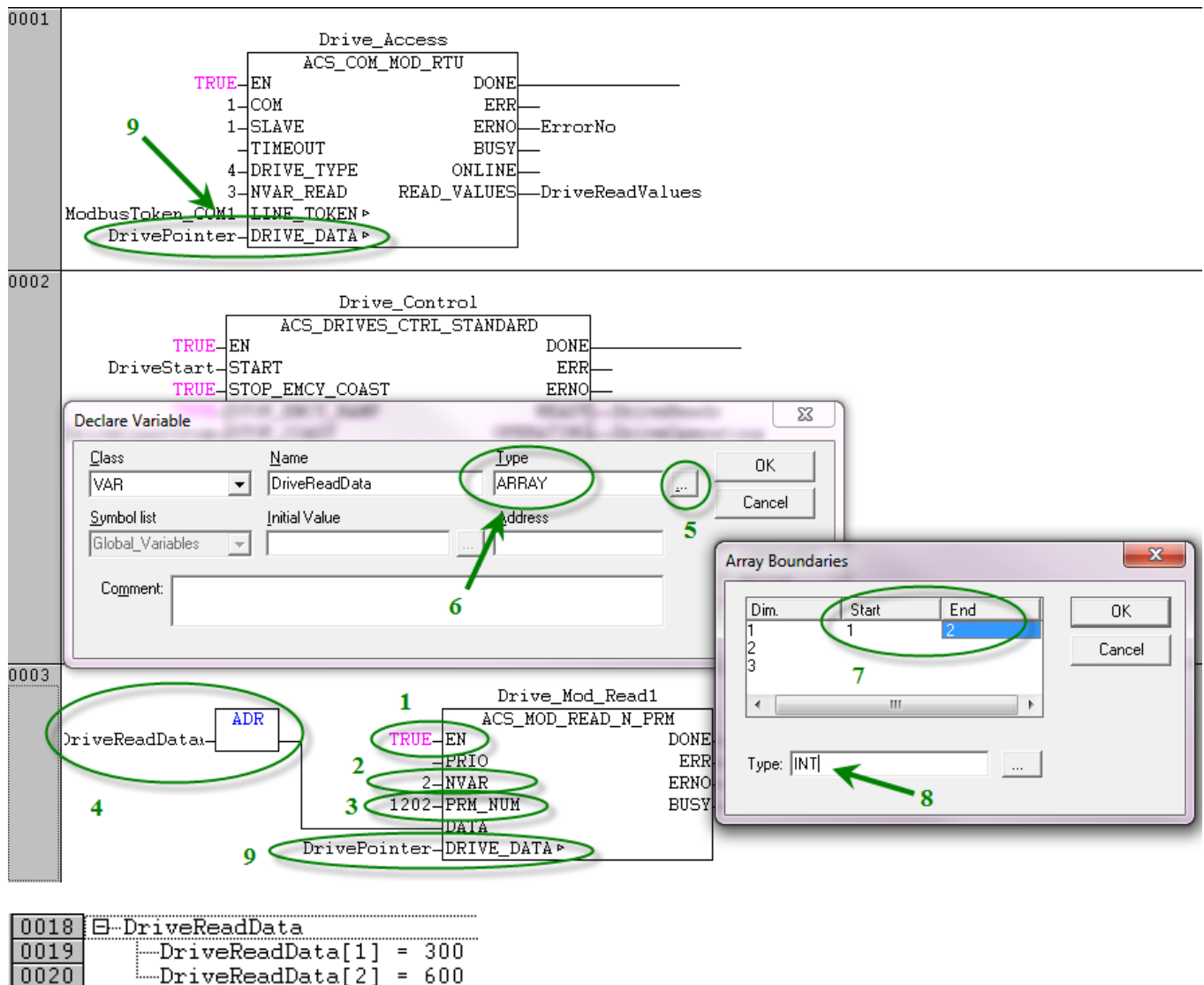
The **ACS\_MOD\_READ\_N\_PRM** block is also shown in the **POUs** list on the left side of the interface.

## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

Set function block inputs according to:

- EN = TRUE (1) → The block will always be enabled.
- PRIO = Left unconnected.
- NVAR = Number of parameters to be read (2) → 2 parameters will be read in this example.
- PRM\_NUM = Start address of parameters to be read (3) → parameters 12.02 and 12.03 will be read in this example.
- DATA is connected via an “ADR” block to a new variable (4). Press the “...” button (5) and choose type “ARRAY” (6), set Start index to 1 and End index to the number of values to be read, 2 in this example (7). Set type to “INT” (8) and press “OK”.
- DRIVE\_DATA = The same variable that is connected to “ACS\_COM\_MOD\_RTU” → “DRIVE\_DATA” (9).

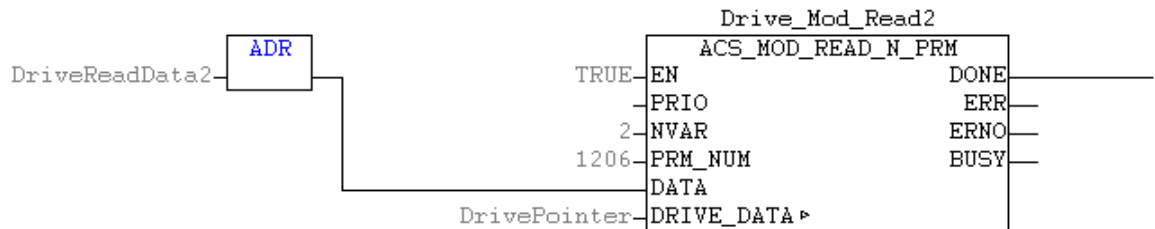
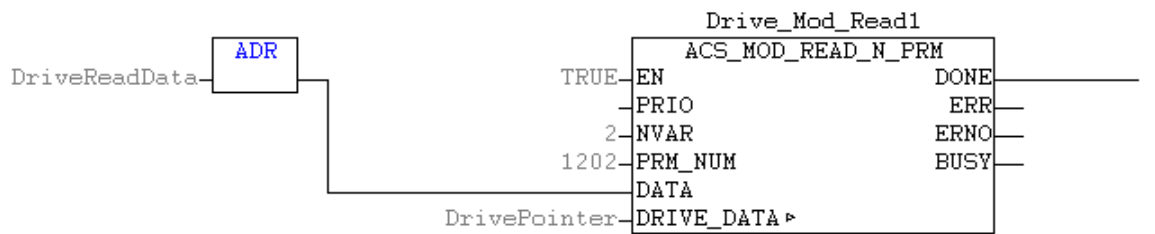


## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

**Note!** Several “ACS\_MOD\_READ\_N\_PRM” blocks can be activated at the same time. See example below.

```
0017 ▣ Drive_Mod_Read1
0018 ▣ DriveReadData
0019   └─ DriveReadData[1] = 350
0020   └─ DriveReadData[2] = 650
0021 ▣ Drive_Mod_Write1
0022 ▣ DriveWriteData
0023   ExecuteWriteData = TRUE
0024 ▣ Drive_Mod_Read2
0025 ▣ DriveReadData2
0026   └─ DriveReadData2[1] = 700
0027   └─ DriveReadData2[2] = 800
0028
```





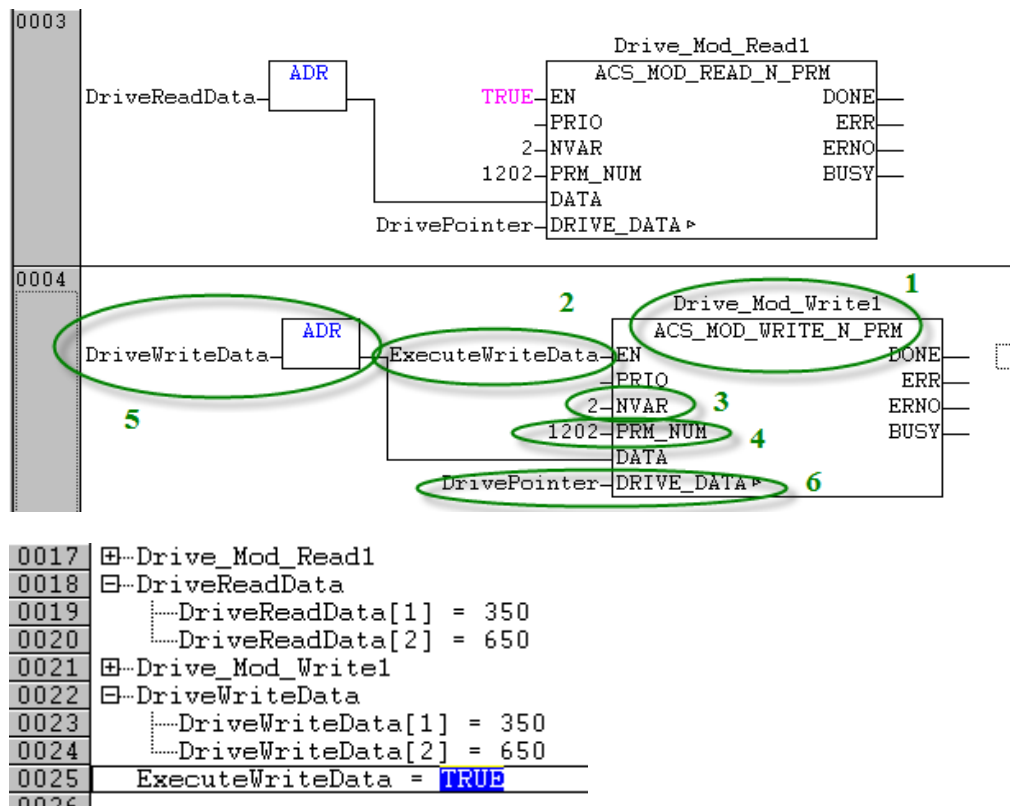
### Example - Write drive data with "ACS\_MOD\_WRITE\_N\_PRM"

Values are written to Modbus register addresses according to user choice. No additional parameter settings in the drive are necessary.

Create a new network in the same program (POU) and add the block "ACS\_MOD\_WRITE\_N\_PRM" (1).

Set function block inputs according to:

- EN is connected to an "execute" variable of type "BOOL" (1). At rising edge the variable is written once.
- PRIO = Left unconnected.
- NVAR = Number of parameters to write (3) → 2 parameters will be written in this example.
- PRM\_NUM = Start address of parameters to be written (4) → parameters 12.02 and 12.03 will be written in this example.
- DATA is connected via an "ADR" block to a new variable (5) of type "ARRAY [1..X] OF INT" according to instructions for the "ACS\_MOD\_READ\_N\_PRM" block.
- DRIVE\_DATA = The same variable that is connected to "ACS\_COM\_MOD\_RTU" → "DRIVE\_DATA" (6).



## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

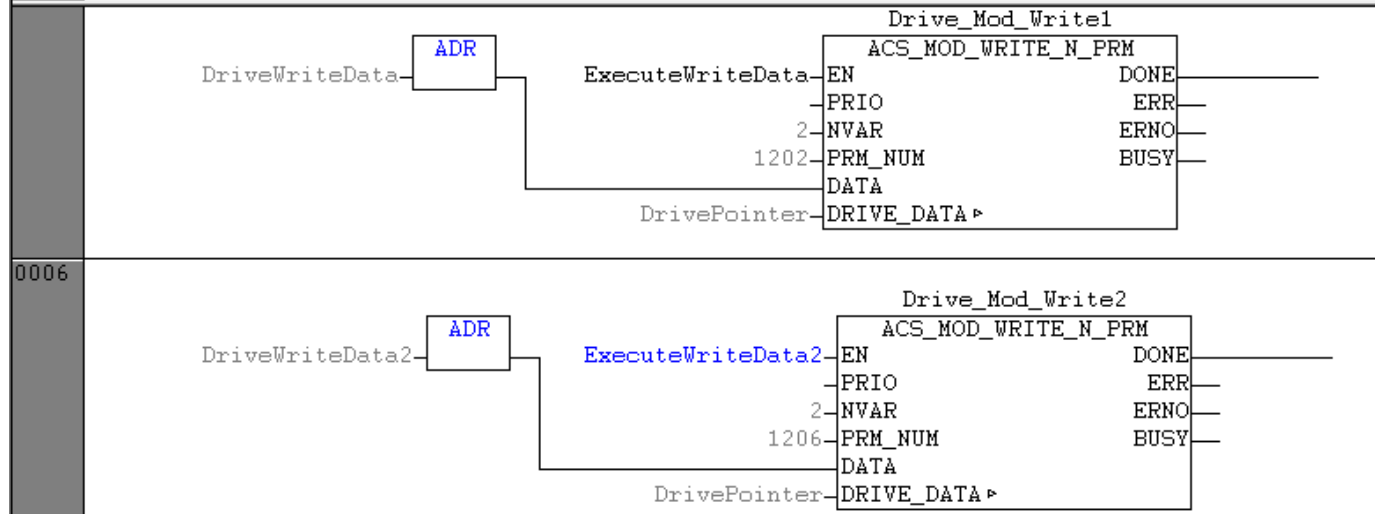
**Note!** Several “ACS\_MOD\_WRITE\_N\_PRM” blocks can be used in the program. See example below.

**Note!** The variable is only written at a rising edge of the EN input.

```

0020 ▢--DriveWriteData
0021   ▢--DriveWriteData[1] = 350
0022   ▢--DriveWriteData[2] = 650
0023   ExecuteWriteData = FALSE
0024 ▢--Drive_Mod_Read2
0025 ▢--DriveReadData2
0026 ▢--Drive_Mod_Write2
0027 ▢--DriveWriteData2
0028   ▢--DriveWriteData2[1] = 750
0029   ▢--DriveWriteData2[2] = 850
0030   ExecuteWriteData2 = TRUE
0031

```

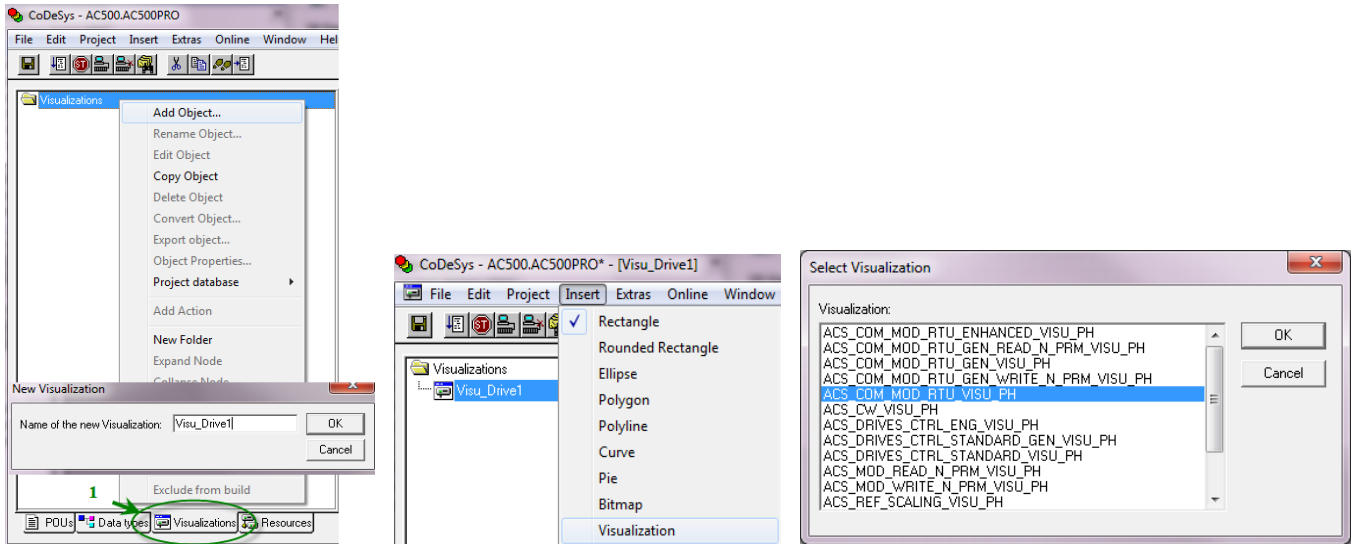


## Quickstart Guide

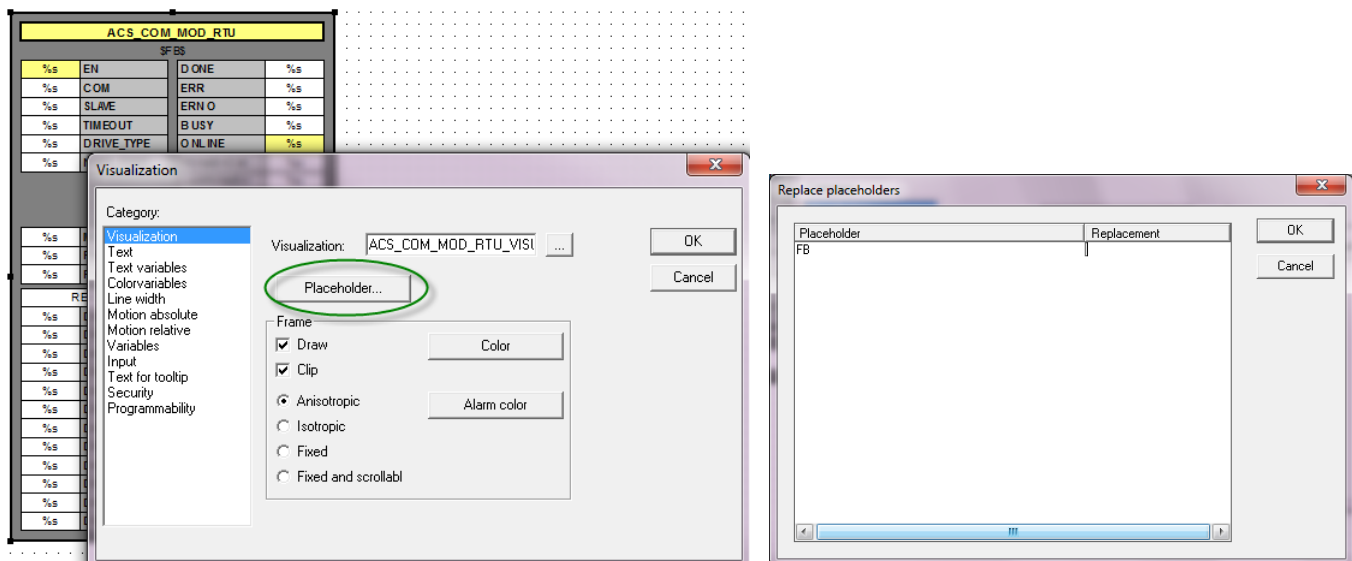
### ABB PLC and drives integration using Modbus RTU

#### Visualizations

Visualizations are **optional** and can be a good way to test the communication between PLC and drive. Go offline (“Logout” from the “Online” menu). Right-click “Visualizations” in the “Visualizations” tab (1) and choose “Add Object”, give the visualization page a suitable name. From the new page, choose “Visualization” from the “Insert” menu and draw a box. Select Visualization “ACS\_COM\_MOD\_RTU\_VISU\_PH”.



Double-click the new visualization object for Settings and click “Placeholder”. Select the “Replacement” field and press F2.



---

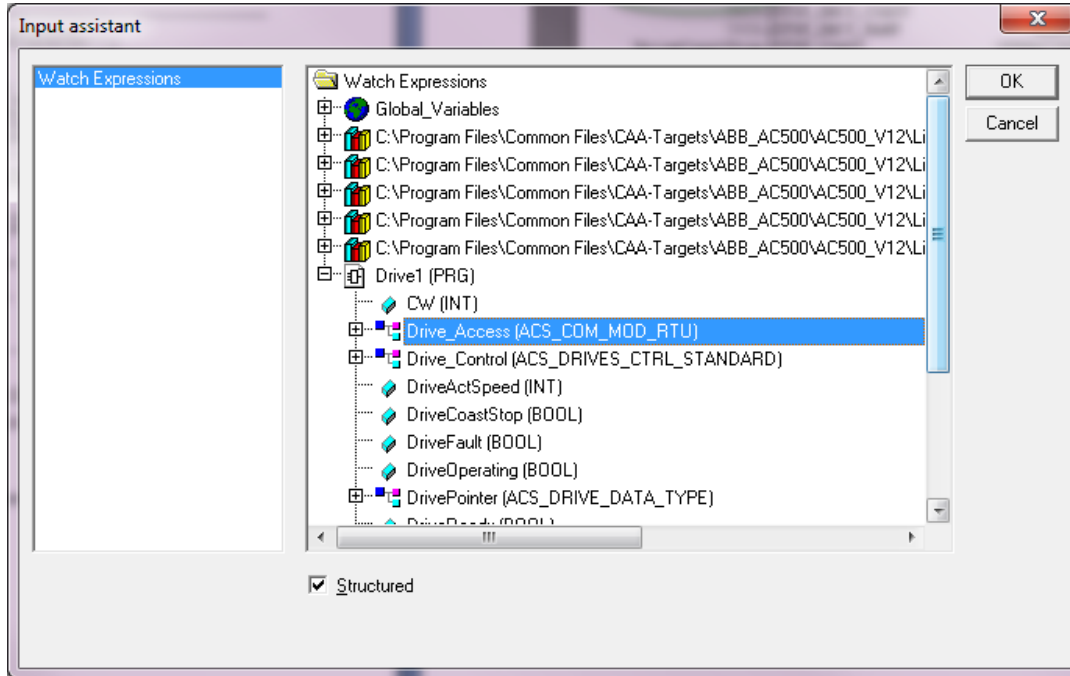
## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

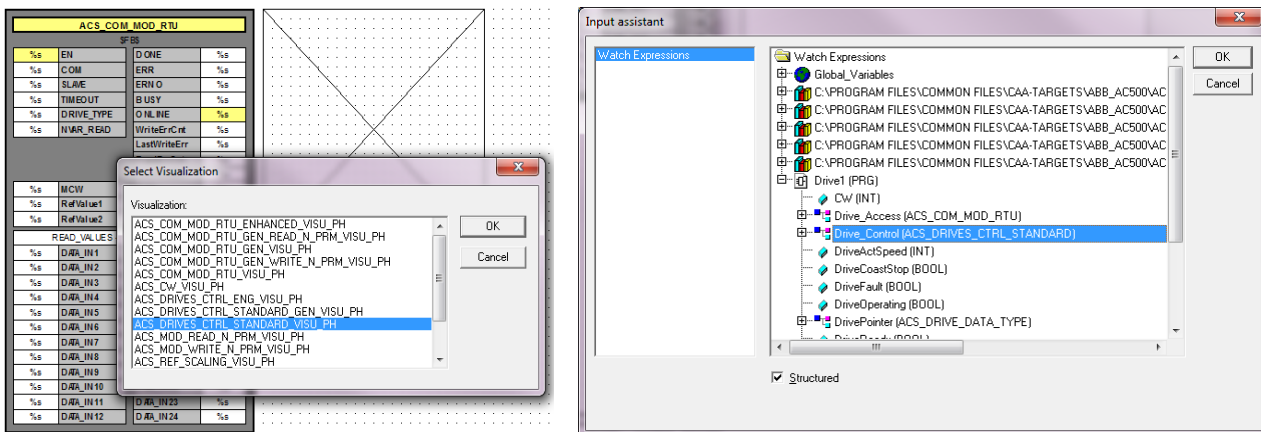
---

To make a connection to the instance of the function block “ACS\_COM\_MOD\_RTU”, choose the instance from actual program (PRG) and click “OK” until all pop-up windows are closed.

**Tip!** Tick the “Structured” box in the Input assistant window for a better overview.



On the same page, create a visualization window for the drive control. Insert → Visualization → Select “ACS\_DRIVES\_CTRL\_STANDARD\_VISU\_PH” → double-click the new object and click “Placeholder” → Press F2 from the “Replacement” field → Choose the instance of function block “ACS\_DRIVES\_CTRL\_STANDARD” → click “OK” until all pop-up windows are closed.



---

## Quickstart Guide

### ABB PLC and drives integration using Modbus RTU

---

Go online again by choosing “Login” from the “Online” menu and then “Run” from the same menu (if the PLC is not already in run mode).

The status of the function block inputs and outputs are now displayed in the visualization window. Block inputs which have not been connected to the function blocks in the program “PRG” can be changed from the visualization window.

**Note!** In this example all block inputs have already been connected to the function blocks, so they cannot be changed from the visualization view.

Some examples of what can be seen in the visualization:

- READY: if the drive is ready to start.
- OPERATING: if the drive is operating (magnetizing the motor).
- TRIPPED: if the drive has an active fault.
- ACT\_SPEED: actual speed value.
- ACT\_SW: actual 16-bit Status Word from the drive.
- USED\_CW: actual 16-bit Control Word sent from the PLC to the drive.
- MESSAGE field: information about the actual state, e.g. “Operation” or “Wait for rising edge of START”.

ACS_COM_MOD_RTU			
Drive1.Drive_Access			
TRUE	EN	DONE	FALSE
1	COM	ERR	FALSE
1	SLAVE	ERNO	0
1000	TIMEOUT	BUSY	TRUE
4	DRIVE_TYPE	ONLINE	TRUE
0	NVAR_READ	WriteErrCnt	0
		LastWriteErr	0
		ReadErrCnt	7
		LastReadErr	8211
1151	MICW	MSW	5943
20000	RefValue1	ActValue1	1500
0	RefValue2	ActValue2	159
READ_VALUES - for embedded modbus			
0	DATA_IN1	DATA_IN13	0
0	DATA_IN2	DATA_IN14	0
0	DATA_IN3	DATA_IN15	0
0	DATA_IN4	DATA_IN16	0
0	DATA_IN5	DATA_IN17	0
0	DATA_IN6	DATA_IN18	0
0	DATA_IN7	DATA_IN19	0
0	DATA_IN8	DATA_IN20	0
0	DATA_IN9	DATA_IN21	0
0	DATA_IN10	DATA_IN22	0
0	DATA_IN11	DATA_IN23	0
0	DATA_IN12	DATA_IN24	0

ACS_DRIVES_CTRL_STANDARD			
Drive1.Drive_Control			
TRUE	EN	DONE	TRUE
TRUE	START	ERR	FALSE
TRUE	EMCY_COAST	ERNO	0
TRUE	EMCY_RAMP	READY	TRUE
FALSE	STOP_COAST	OPERATING	TRUE
FALSE	RESET	TRIPPED	FALSE
		ALARM	FALSE
		EXT_RUN_EN	TRUE
		LOCAL_CTRL	FALSE
FALSE	EXT_CTRL_LOC	EXT_CTRL_LOC	FALSE
20000	SPEED_REF	ACT_SPEED	1500
0	REF_VALUE2	ACT_VALUE2	159
		ACT_SW	5943
		USED_CW	1151
MESSAGE		Operation	

---

## Quickstart Guide

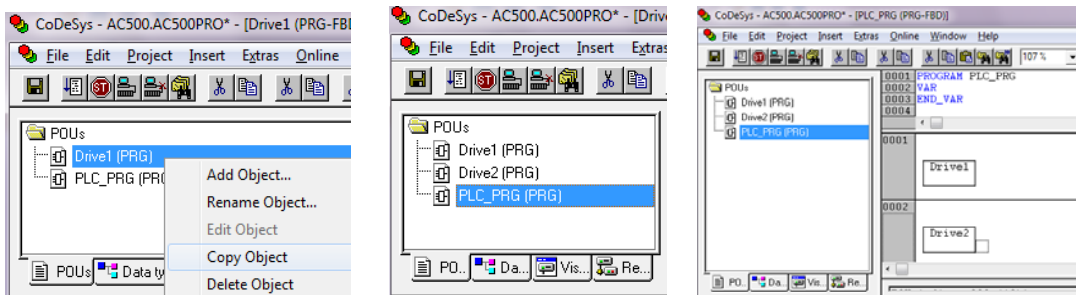
### ABB PLC and drives integration using Modbus RTU

---

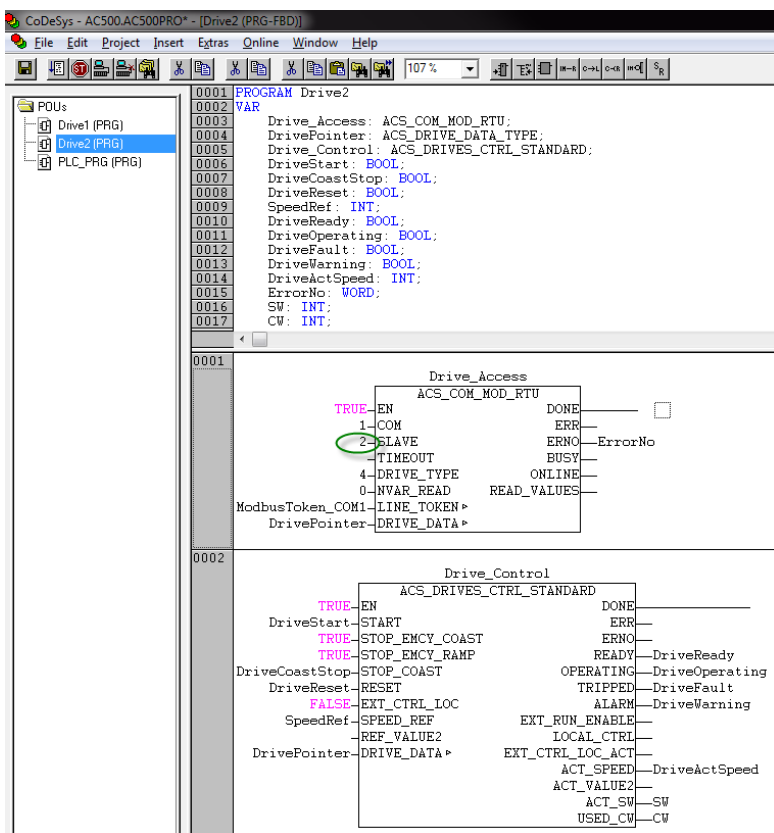
Go offline by choosing “Logout” from the “Online” menu.

#### Add more drives (optional)

In the CODESYS “POUs” tab, right-click the program for your earlier drive and choose “Copy Object”, give the new program a suitable name and double-click “PLC\_PRG”. Add the new program to the main program by copying Network 0001 to Network 0002 and rename the block in Network 0002 to according to your new program.



Double-click and open the new program. Change “SLAVE” number of the “ACS\_COM\_MOD\_RTU” block according to the Modbus RTU node address of actual drive.



Continue to add as many drives as you need. If you want to create visualization pages/objects for the additional drives, make sure that you connect the Placeholders to the function block instances for the correct program (PRG), “Drive2” in this example.

### Actualization rate

Be aware that on a Modbus RTU line only one Modbus job is performed at a time. That means that the update rate of the variables increases with the number of active Modbus blocks, such as ACS\_COM\_MOD\_RTU, ACS\_MOD\_READ\_N\_PRM of ACS\_MOD\_WRITE\_N\_PRM.

The ACS\_COM\_MOD\_RTU function block automatically reads the actual data all the time and executes a write job each time the Control Word or the reference values (speed or torque) have changed.

### Other useful documentation

- CODESYS Help (Contents → Target system → AC500 / S500 → ACS Drives Libraries)
- User's manual Modbus Adapter Module FMBA-01 [3AFE68586704]
- User's manual Modbus Adapter Module RMBA-01 [3AFE 64498851]
- Modbus Protocol Manual RS-485 Adapter Module FSMA-01 [3AUA0000044530]
- User's manual ACS355 drives [3AUA0000066143]
- ACS850 Firmware Manual Standard Control Program [3AUA0000045497]
- ACQ810 Firmware Manual Standard Pump Control Program [3AUA0000055144]
- Firmware manual ACS880 primary control program [3AUA0000085967]
- ACSM1 Firmware Manual Speed and Torque Control [3AFE68848270]
- ACSM1 Firmware Manual Motion Control [3AFE68848270]
- User's Manual ACS550-01 Drives [3AUA0000001418]
- Firmware Manual ACS800 Standard Control Program [3AFE64527592]

### Revision

Rev	Page (P) Chap (C)	Description	Date
A	-	New document	2013-02-01
B		Update (C) Drive configuration	2013-03-22

For more information please contact your local ABB  
representative or visit:  
**[www.abb.com/drives](http://www.abb.com/drives)**  
**[www.abb.com/drivespartners](http://www.abb.com/drivespartners)**

© Copyright 2012 ABB. All rights reserved.  
Specifications subject to change without notice.