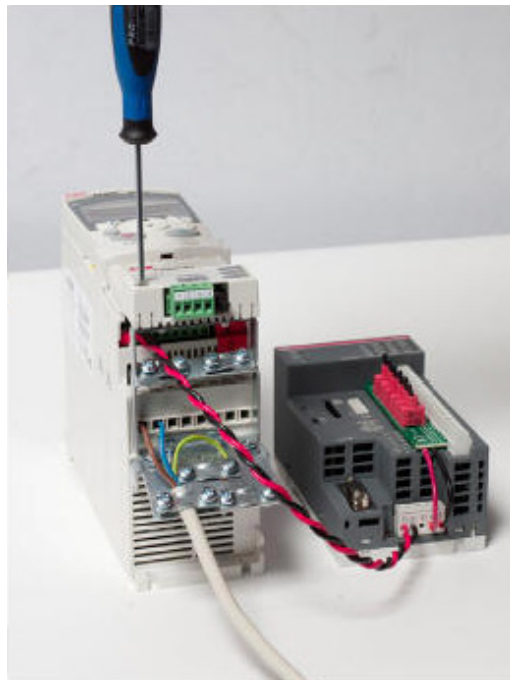


Application guide

ACS355 and AC500-eCo



List of related manuals

Drive and PLC hardware manuals and guides	Code (English)
<i>ACS355 user's manual</i>	3AUA0000066143 2)
<i>AC500-eCo and ACS355 quick installation guide</i>	2CDC125145M0201 1)
<i>System description AC500</i>	2CDC125015M0201 2)
<i>PM554 and PM564 installation instructions</i>	2CDC125122M6801 2)
<i>AC500 online help</i>	3)

Option manuals and guides

<i>FMBA-01 Modbus adapter module user's manual</i>	3AFE68586704 1)
--	---------------------------------

- 1) Delivered as a printed copy with the starter kit.
- 2) Delivered on the SD memory card in the folder \PS553-DRIVES\Documentation.
- 3) Can be accessed through the CoDeSys program. See [Using the online help in CoDeSys](#) on page 15.

Application guide

ACS355 and AC500-eCo

Table of contents



Table of contents

List of related manuals	2
1. About the manual	
What this chapter contains	7
Starter kit overview	7
Compatibility	7
Safety instructions	7
Reader	7
Contents	8
Related manuals	8
2. Configuration	
What this chapter contains	9
ACS355 drive configuration	10
Installing ABB Control Builder AC500	11
Installing PS553-DRIVES libraries	11
Configuring communication parameters	12
Opening the example project	12
Installing the TK504 cable	12
Checking the COM port of TK504	13
Setting new communication parameter to COMx for TK504	13
Connecting to the PLC	13
3. Customization	
What this chapter contains	15
Getting help	15
Using the online help in CoDeSys	15
Getting started AC500	15
AC500 ACS Drives Libraries topics	15
Getting help on a specific function block	16
Configuring the Modbus settings of the AC500	16
Configuring the slave address of the drive	17
Adding another drive	17
Copying the program for another drive	18
Information on LineCom1 as a global variable	18
Creating a visualization for a copied drive program	19
Using visualizations	20
Controlling the drive with the visualization	20
Configuring parameters to be read from the drive	21
Configuration example	21
Using the read parameters in the program	22
Word-to-integer conversion for variables with negative values	22
4. Function blocks and program structure	
What this chapter contains	23
Function block programming overview	23



Program structure	24
Main program	24
PRG_Drive1 program	24
DriveRef	25
Main components of the example program	25
Modbus RTU communication function block (FB_COM)	25
Basic control block (FB_BASIC CONTROL)	25
Speed reference generation	26
Other control blocks	26
ACS_CTRL_ABB_DRV_PROFILE	26
ACS_DRIVES_CTRL_ENG	27





1

About the manual

What this chapter contains

The chapter describes the contents of the manual. It also contains information on the compatibility, safety and intended reader.

Starter kit overview

The ACS355 and AC500-eCo starter kit is designed for controlling up to seven ACS355 drives connected through a Modbus link. The maximum total length of the Modbus link is 50 meters (160 feet).

Compatibility

This manual is intended to be used with the ACS355 and AC500-eCo starter kit.

Safety instructions

Follow all safety instructions delivered with the starter kit:

- Read the complete safety instructions for the ACS355 drive before you install, commission or use the drive. The complete safety instructions are given at the beginning of the *ACS355 user's manual* (3AUA0000066143 [English]).
- Read all safety instructions of the AC500-eCo PLC. See *System description AC500* (2CDC125015M0201 [English]) or the online help in CoDeSys (Help > Contents > Target System > AC500 / S500 > Introduction > Overview > Regulations).

Reader

The manual is intended for people responsible for configuring and using the starter kit.

Contents

The manual consists of the following chapters:

- [Configuration](#) describes how to install and configure the software needed for using the ACS355 drive with the AC500-eCo PLC.
- [Customization](#) describes optional configuration and customization options.
- [Function blocks and program structure](#) describes the principles of ACS Drives Libraries function blocks and the structure of the example program.

Related manuals

In addition to this manual, the delivery includes the following manuals:

- *Quick installation guide* which describes the mechanical and electrical installation of the starter kit.
- *ACS355 user's manual* which describes the ACS355 drive.
- *System description AC500* which contains the safety instructions for the AC500-eCo PLC.
- *PM554 and PM564 installation instructions* which describes the installation of the AC500-eCo PLC.
- The online help in the CoDeSys program (see [Using the online help in CoDeSys](#) on page 15.)
- *FMBA-01 user's manual* which describes the FMBA-01 Modbus adapter module.

For a complete list of related manuals, see the inside of the front cover.



2

Configuration

What this chapter contains
















This chapter describes how to install ABB PS501 Control Builder on your computer, and how to configure the ACS355 drive and AC500-eCo PLC for use.

The instructions in this chapter are intended to be used after the mechanical and electrical installation has been completed, as explained in *AC500-eCo and ACS355 quick installation guide*.

Instructions for optional configuration and customization are given in chapter [Customization](#).

ACS355 drive configuration

The AC500 Modbus application macro makes the necessary changes to parameter default values for use in the starter kit. To activate the macro, set parameter 9902 APPLIC MACRO to AC500 MODBUS (10) as shown below. The macro is available in ACS355 drives with software version 503C or above.

Step	Action	Display
1.	Go to the Main menu by pressing  if you are in the Output mode, otherwise by pressing repeatedly  until you get to the Main menu.	<pre> LOC MAIN MENU 1 PARAMETERS ASSISTANTS CHANGED PAR EXIT 00:00 ENTER </pre>
2.	Go to the Parameters mode by selecting PARAMETERS on the menu with keys  and  , and pressing  .	<pre> LOC PAR GROUPS 01 01 OPERATING DATA 03 FB ACTUAL SIGNALS 04 FAULT HISTORY 10 START/STOP/DIR 11 REFERENCE SELECT EXIT 00:00 SEL </pre>
3.	Select the appropriate parameter group with keys  and  . Press  .	<pre> LOC PAR GROUPS 99 99 START-UP DATA 01 OPERATING DATA 03 FB ACTUAL SIGNALS 04 FAULT HISTORY 10 START/STOP/DIR EXIT 00:00 SEL </pre> <pre> LOC PARAMETERS 9901 LANGUAGE ENGLISH 9902 APPLIC MACRO 9903 MOTOR TYPE 9904 MOTOR CTRL MODE EXIT 00:00 EDIT </pre>
4.	Select the appropriate parameter with keys  and  . The current value of the parameter is shown below the selected parameter. Press  .	<pre> LOC PARAMETERS 9901 LANGUAGE 9902 APPLIC MACRO ABB STANDARD 9903 MOTOR TYPE 9904 MOTOR CTRL MODE EXIT 00:00 EDIT </pre> <pre> LOC PAR EDIT 9902 APPLIC MACRO ABB STANDARD [1] CANCEL 00:00 SAVE </pre>
5.	Specify a new value for the parameter with keys  and  . Pressing the key once increments or decrements the value. Holding the key down changes the value faster. Pressing the keys simultaneously replaces the displayed value with the default value.	<pre> LOC PAR EDIT 9902 APPLIC MACRO AC500 MODBUS [10] CANCEL 00:00 SAVE </pre>
6.	<ul style="list-style-type: none"> To save the new value, press . To cancel the new value and keep the original, press . 	<pre> LOC PARAMETERS 9901 LANGUAGE 9902 APPLIC MACRO AC500 MODBUS 9903 MOTOR TYPE 9904 MOTOR CTRL MODE EXIT 00:00 EDIT </pre>

The AC500 Modbus application macro default drive parameters correspond to the ABB standard macro for ACS355, with the following changes:

Parameter	Name	Value
1001	EXT1 COMMANDS	COMM (10)
1102	EXT1/EXT2 SEL	COMM (8)
1103	REF1 SEL	COMM (8)
1604	FAULT RESET SEL	COMM (8)

2201	ACC/DEC 1/2 SEL	NOT SEL (0)
3018	COMM FAULT FUNC	FAULT (1)
5302	EFB STATION ID	2
5303	EFB BAUD RATE	19.2 kbit/s (192)
5304	EFB PARITY	8 NONE 1 (1)
5305	EFB CTRL PROFILE	ABB DRV FULL (2)
5310	EFB PAR 10	101
5311	EFB PAR 11	303
5312	EFB PAR 12	305
9802	COMM PROT SEL	STD MODBUS (1)

Installing ABB Control Builder AC500

This section describes how to install the Control Builder and a driver for the TK503/504 cable.

Note: For more information on the cable driver installation, see [\CD_AC500\Driver\TK503_TK504\TK503_TK504_Driver_Installation.pdf](#) on the installation CD.

1. Insert the installation CD of ABB PS501 Control Builder.

The CD automatically starts the installation program. A new window opens.

2. Click **Installation PS501**.

The installation begins.

3. Follow the steps in the installation wizard.

Once the installation is finished, you return to the main menu.

4. Click **Tools**.

5. Click **TK503 + TK504 cable driver**.

The TK503_TK504 programming cable Driver Installer window opens.

6. Click **Install**.

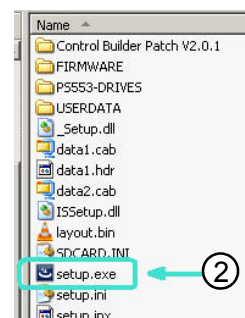
The installation begins.

Installing PS553-DRIVES libraries

Note: ABB PS501 Control Builder must be installed first.

1. Remove the SD memory card from the AC500-eCo and insert it in the memory card reader of the PC.
2. In the root directory of the SD memory card, open setup.exe.
3. Follow the steps of the setup wizard.

Libraries, examples and the online help are copied to the hard drive of the PC.



12 Configuration

Configuring communication parameters

The below instructions demonstrate how to configure communication parameters with the example project.

■ Opening the example project

1. In the Windows Start menu, open the CoDeSys programming tool by clicking **Programs > 3S Software > CoDeSys V2.3 > CoDeSys V2.3**.

The CoDesys program opens.

2. Click **File > Open**.
3. Browse to the folder **C:\Program Files\ABB\ABB Configurator\Projects\Examples\Drives_PS553-DRIVES\ACS355_PM554_ModbusRTU** and open **ACS355_PM554_ModbusRTU.pro**.

A dialog opens, asking if you want to save the current (empty) project.

4. Click **No**.

The project opens and the PLC_VISU window and the PRG_DRIVE1 window are displayed.

The screenshot shows the CoDeSys software interface. The main window is divided into several panes:

- PLC_VISU:** Contains two tables.

ACS3XX_DRIVES_CTRL_BASIC			
%s	EN	DONE	%s
%s	START	ERR	%s
%s	STOP_COAST	ERNO	%s
%s	RESET	SCALING_DONE	%s
		OPERATING	%s
		TRIPPED	%s
		ALARM	%s
%s	SPEED_REF	ACT_SPEED	%s

ACS3XX_COM_MOD_RTU			
%s	EN	DONE	%s
%s	COM	ERR	%s
%s	TIMOUT	ERNO	%s
%s	LINE	BUSY	%s
%s	DRIVE_TYPE	ONLINE	%s
	WriteErrCnt		%s
	ListWriteErrno		%s
	ReadErrCnt		%s
	ListReadErrno		%s
%s	NVAR_READ	SV	%s
	ActValu1		%s
	ActValu2		%s
	ActValu3		%s
	ActValu4		%s
	ActValu5		%s
	ActValu6		%s
	ActValu7		%s
	ActValu8		%s
- PRG_DRIVE1 (PRG_FB0):** Contains a variable declaration:


```

0001 PROGRAM PRG_Drive1
0002 VAR
0003   FB_COM: ACS3XX_COM_MOD_RTU; (* Communicat
0004   DriveRef: ACS_DRIVE_DATA_TYPE; (* Drive referen
0005   FB_BASIC_CONTROL: ACS3XX_DRIVES_CTRL_BASIC; (* Basic control
0006
0007   rRefSpeed: REAL; (* speed reference in rpm or 0.1Hz, depen
0008   rActSpeed: REAL; (* actual speed in rpm if ACS355 Parame
0009   xOnlineOK: BOOL; (* Online - connection established; one m
0010   awACT_PARAMETERS: ARRAY[1..9] OF WORD := $0(); (* actual Param
0011
0012 END_VAR
      
```
- 0001:** modbus RTU communication function block:


```

          FB_COM
          ACS3XX_COM_MOD_RTU
          TRUE -> EN
          1 -> COM
          2 -> SLAVE
          1000 -> TIMEOUT
          ACS_DRIVE_ACS355 -> DRIVE_TYPE
          3 -> NVAR_READ
          LineCom1 -> LINE_TOKEN >
          DriveRef -> DRIVE_DATA >
          DONE
          ERR
          ERNO
          BUSY
          ONLINE -> xOnlineOK
          READ_VALLE -> awACT_PARAMETERS
      
```
- 0002:** control function block, including scaling and initial reset:


```

          FB_BASIC_CONTROL
          ACS3XX_DRIVES_CTRL_BASIC
          xOnlineOK -> EN
          DL_START -> START
          DL_COAST_STOP -> STOP_COAST
          DL_RESET -> RESET
          rRefSpeed -> SPEED_REF
          DriveRef -> DRIVE_DATA >
          DONE
          ERR
          ERNO
          SCALING_DONE
          OPERATING
          TRIPPED
          ALARM
          ACT_SPEED -> ActSpeed
      
```

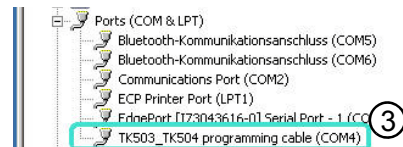
■ Installing the TK504 cable

Plug the TK504 cable in a USB port on the PC.

An installation wizard for the TK504 cable opens. Follow the instructions in the wizard to automatically install the cable.

■ Checking the COM port of TK504

1. In Windows Start menu, go to **Settings > Control Panel > System**.
2. Select the **Hardware** tab and click **Device Manager**.
3. Expand **Ports (COM & LPT)** and check to which COM port **TK503_504 programming cable** is assigned. The COM port is shown in parentheses.



■ Setting new communication parameter to COMx for TK504

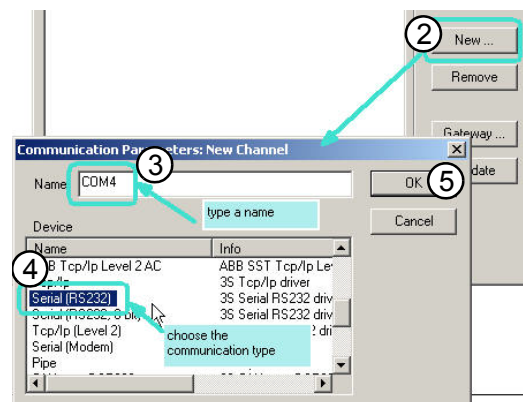
1. In CoDeSys, click **Online > Communication Parameters....**

The Communication Parameters window opens.

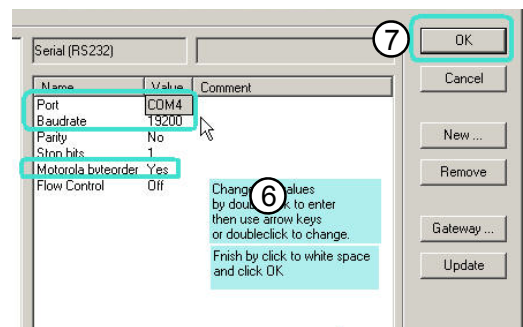
2. Click **New**

The Communication Parameters: New Channel window opens.

3. In the **Name** field, enter a name corresponding the COM port, such as "COM7".
4. In the **Device** field, select **Serial (RS232)**.
5. Click **OK** to close the window.



6. Change the following values:
 - Port: [the corresponding COM port]
 - Baudrate: **19200**
 - Motorola byteorder: **Yes**
7. Click **OK** to exit.



■ Connecting to the PLC

Click **Online > Login**.

14 Configuration

The status of the function blocks is displayed.

The screenshot displays the CODESYS environment for a project named 'ACS355_PM554_MotibusRTU.pro'. The interface is divided into several panes:

- Left Pane (POUs):** Shows a tree view with 'PLC_PRG (PRG)' and 'PRG_Drive1 (PRG)' selected.
- Top Middle Pane (PLC_VISU):** Displays two data tables for function blocks.

ACS3XX_DRIVES_CTRL_BASIC			
PRG_Drive1FB_BASIC_CONTROL			
TRUE	EN	DONE	FALSE
FALSE	SWRT	ERR	FALSE
FALSE	STOP_COAST	ERNO	0
FALSE	RESET	SCALING_DONE	TRUE
		OPERATING	FALSE
		TRIPPED	FALSE
	ALARM		FALSE
0.00	SPEED_REF	ACT_SPEED	0.00

ACS3XX_COM_MOD_RTU			
PRG_Drive1FB_COM			
TRUE	EN	DONE	FALSE
1	COM	ERR	FALSE
1000	TIMOUT	ERNO	0
2	SLAVE	BUSY	FALSE
4	DRIVE_TYPE	ONLINE	TRUE
	WriteErrCnt		0
	LastWriteErrno		0
	ReadErrCnt		0
	LastReadErrno		0
0	INVAR_READ	SW	#557
	ActValue1		0
	ActValue2		19
	ActValue3		0
	ActValue4		0
	ActValue5		0
	ActValue6		0
	ActValue7		0
	ActValue8		0
- Right Pane (PRG_Drive1 (PRG-FBD)):** Shows the ladder logic diagram. It includes:
 - Network 0001: Declaration of function blocks.
 - Network 0002: Declaration of 'DriveRef'.
 - Network 0003: Declaration of 'FB_BASIC_CONTROL'.
 - Network 0004: Declaration of 'rRefSpeed = 0'.
 - Network 0005: Declaration of 'rActSpeed = 0'.
 - Network 0006: Declaration of 'xOnlineOK = TRUE'.
 - Network 0007: Declaration of 'awACT_PARAMETERS'.
 - Network 0008: Declaration of 'awACT_PARAMETERS'.
 - Network 0009: Declaration of 'awACT_PARAMETERS'.
 - Network 0010: Declaration of 'awACT_PARAMETERS'.
 - Network 0011: Declaration of 'awACT_PARAMETERS'.
 - Network 0012: Declaration of 'awACT_PARAMETERS'.
 - Network 0013: Declaration of 'awACT_PARAMETERS'.
 - Network 0014: Declaration of 'awACT_PARAMETERS'.
 - Network 0015: Declaration of 'awACT_PARAMETERS'.
 - Network 0016: Declaration of 'awACT_PARAMETERS'.
 - Network 0017: Declaration of 'awACT_PARAMETERS'.
 - Network 0018: Declaration of 'awACT_PARAMETERS'.
 - Network 0019: Declaration of 'awACT_PARAMETERS'.
 - Network 0020: Declaration of 'awACT_PARAMETERS'.
 - Network 0021: Declaration of 'awACT_PARAMETERS'.
 - Network 0022: Declaration of 'awACT_PARAMETERS'.
 - Network 0023: Declaration of 'awACT_PARAMETERS'.
 - Network 0024: Declaration of 'awACT_PARAMETERS'.
 - Network 0025: Declaration of 'awACT_PARAMETERS'.
 - Network 0026: Declaration of 'awACT_PARAMETERS'.
 - Network 0027: Declaration of 'awACT_PARAMETERS'.
 - Network 0028: Declaration of 'awACT_PARAMETERS'.
 - Network 0029: Declaration of 'awACT_PARAMETERS'.
 - Network 0030: Declaration of 'awACT_PARAMETERS'.
 - Network 0031: Declaration of 'awACT_PARAMETERS'.
 - Network 0032: Declaration of 'awACT_PARAMETERS'.
 - Network 0033: Declaration of 'awACT_PARAMETERS'.
 - Network 0034: Declaration of 'awACT_PARAMETERS'.
 - Network 0035: Declaration of 'awACT_PARAMETERS'.
 - Network 0036: Declaration of 'awACT_PARAMETERS'.
 - Network 0037: Declaration of 'awACT_PARAMETERS'.
 - Network 0038: Declaration of 'awACT_PARAMETERS'.
 - Network 0039: Declaration of 'awACT_PARAMETERS'.
 - Network 0040: Declaration of 'awACT_PARAMETERS'.
 - Network 0041: Declaration of 'awACT_PARAMETERS'.
 - Network 0042: Declaration of 'awACT_PARAMETERS'.
 - Network 0043: Declaration of 'awACT_PARAMETERS'.
 - Network 0044: Declaration of 'awACT_PARAMETERS'.
 - Network 0045: Declaration of 'awACT_PARAMETERS'.
 - Network 0046: Declaration of 'awACT_PARAMETERS'.
 - Network 0047: Declaration of 'awACT_PARAMETERS'.
 - Network 0048: Declaration of 'awACT_PARAMETERS'.
 - Network 0049: Declaration of 'awACT_PARAMETERS'.
 - Network 0050: Declaration of 'awACT_PARAMETERS'.
 - Network 0051: Declaration of 'awACT_PARAMETERS'.
 - Network 0052: Declaration of 'awACT_PARAMETERS'.
 - Network 0053: Declaration of 'awACT_PARAMETERS'.
 - Network 0054: Declaration of 'awACT_PARAMETERS'.
 - Network 0055: Declaration of 'awACT_PARAMETERS'.
 - Network 0056: Declaration of 'awACT_PARAMETERS'.
 - Network 0057: Declaration of 'awACT_PARAMETERS'.
 - Network 0058: Declaration of 'awACT_PARAMETERS'.
 - Network 0059: Declaration of 'awACT_PARAMETERS'.
 - Network 0060: Declaration of 'awACT_PARAMETERS'.
 - Network 0061: Declaration of 'awACT_PARAMETERS'.
 - Network 0062: Declaration of 'awACT_PARAMETERS'.
 - Network 0063: Declaration of 'awACT_PARAMETERS'.
 - Network 0064: Declaration of 'awACT_PARAMETERS'.
 - Network 0065: Declaration of 'awACT_PARAMETERS'.
 - Network 0066: Declaration of 'awACT_PARAMETERS'.
 - Network 0067: Declaration of 'awACT_PARAMETERS'.
 - Network 0068: Declaration of 'awACT_PARAMETERS'.
 - Network 0069: Declaration of 'awACT_PARAMETERS'.
 - Network 0070: Declaration of 'awACT_PARAMETERS'.
 - Network 0071: Declaration of 'awACT_PARAMETERS'.
 - Network 0072: Declaration of 'awACT_PARAMETERS'.
 - Network 0073: Declaration of 'awACT_PARAMETERS'.
 - Network 0074: Declaration of 'awACT_PARAMETERS'.
 - Network 0075: Declaration of 'awACT_PARAMETERS'.
 - Network 0076: Declaration of 'awACT_PARAMETERS'.
 - Network 0077: Declaration of 'awACT_PARAMETERS'.
 - Network 0078: Declaration of 'awACT_PARAMETERS'.
 - Network 0079: Declaration of 'awACT_PARAMETERS'.
 - Network 0080: Declaration of 'awACT_PARAMETERS'.
 - Network 0081: Declaration of 'awACT_PARAMETERS'.
 - Network 0082: Declaration of 'awACT_PARAMETERS'.
 - Network 0083: Declaration of 'awACT_PARAMETERS'.
 - Network 0084: Declaration of 'awACT_PARAMETERS'.
 - Network 0085: Declaration of 'awACT_PARAMETERS'.
 - Network 0086: Declaration of 'awACT_PARAMETERS'.
 - Network 0087: Declaration of 'awACT_PARAMETERS'.
 - Network 0088: Declaration of 'awACT_PARAMETERS'.
 - Network 0089: Declaration of 'awACT_PARAMETERS'.
 - Network 0090: Declaration of 'awACT_PARAMETERS'.
 - Network 0091: Declaration of 'awACT_PARAMETERS'.
 - Network 0092: Declaration of 'awACT_PARAMETERS'.
 - Network 0093: Declaration of 'awACT_PARAMETERS'.
 - Network 0094: Declaration of 'awACT_PARAMETERS'.
 - Network 0095: Declaration of 'awACT_PARAMETERS'.
 - Network 0096: Declaration of 'awACT_PARAMETERS'.
 - Network 0097: Declaration of 'awACT_PARAMETERS'.
 - Network 0098: Declaration of 'awACT_PARAMETERS'.
 - Network 0099: Declaration of 'awACT_PARAMETERS'.
 - Network 0100: Declaration of 'awACT_PARAMETERS'.
- Bottom Pane:** Shows the status of the drive control system. The status is 'ONLINE: COM4 [SIM] RUNNING [OP] FORCE [OV] READ'.

3

Customization

What this chapter contains

This chapter describes how to add more drives, customize the program further in CoDeSys and work with visualizations.

Getting help

■ Using the online help in CoDeSys

To open the online help, click **Help > Contents....**

See the sections below for more detailed information on AC500 help topics.

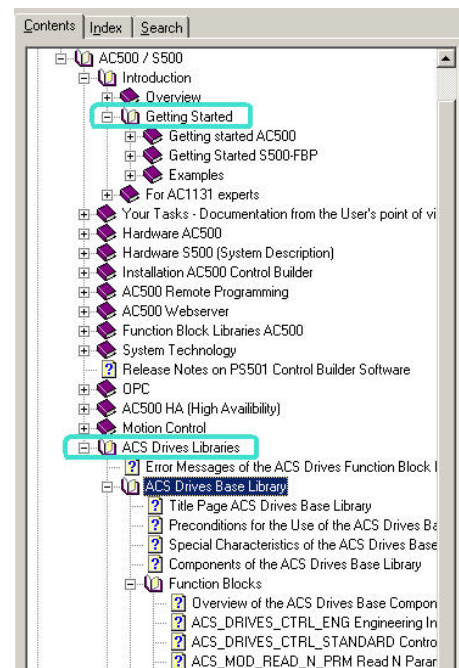
Getting started AC500

In the **Contents** tab, expand **Target System > AC500 / S500 > Introduction > Getting Started** and open **Getting started AC500**.

AC500 ACS Drives Libraries topics

In the **Contents** tab, expand **Target System > AC500 / S500 > ACS Drives Libraries**.

Under ACS Drives Libraries you can see the help topics related to the ACS Drives Libraries.

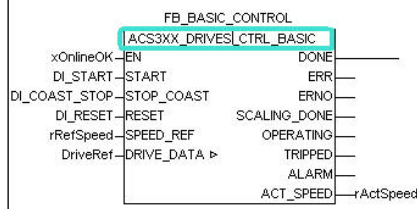


16 Customization

Getting help on a specific function block

In CoDeSys, select the title of the function block and press F1.

The library view also contains information on the function blocks and their inputs and outputs. Press Alt + Enter to enter the library view.



Configuring the Modbus settings of the AC500

You can change the Modbus settings, but they have to be identical in the drive and the AC500.

1. In CoDeSys, in the left pane, select the **Resources** tab.
2. Click **PLC Configuration**.
3. In the middle pane, under AC500, expand **Interfaces[FIX]**.
4. Click **COM1 - MODBUS[SLOT]**.
5. In the right pane (**Module parameters**), configure the following parameters:

Name	Value
RTS control	telegram
Baudrate	(same as in the drive)
Parity	(same as in the drive)
Data bits	(same as in the drive)
Stop bits	(same as in the drive)
Operation mode	Master
Address	0

Note: For AC500 PLCs with firmware version 2.0 or higher, use ABB Control Builder to set up the bus parameters. To open ABB Control Builder, go to the Windows Start menu and open **Programs > ABB > ABB Control Builder**.

Configuring the slave address of the drive

The default slave address of the drive is 2, but if several drives are used, the address must be unique in each drive.

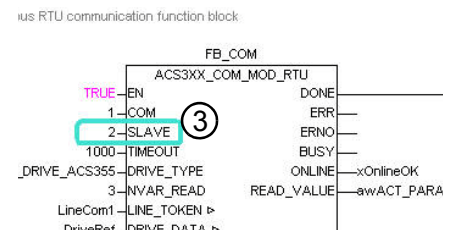
On the drive side, you can choose the slave address in ACS3xx drives with parameter 53.02 EFB STATION ID. You can use the control panel for changing the parameter value. For instructions, see chapter *Control panels* in *ACS355 user's manual* (3AUA0000066143 [English]).

To configure the slave address in CoDeSys, follow the instructions below:

1. In CoDeSys, in the left pane, select the **POUs** tab.
2. Click **PRG_Drive1 (PRG)**.
3. In the **modbus RTU communication function block**, change **SLAVE** to match the Modbus slave number.

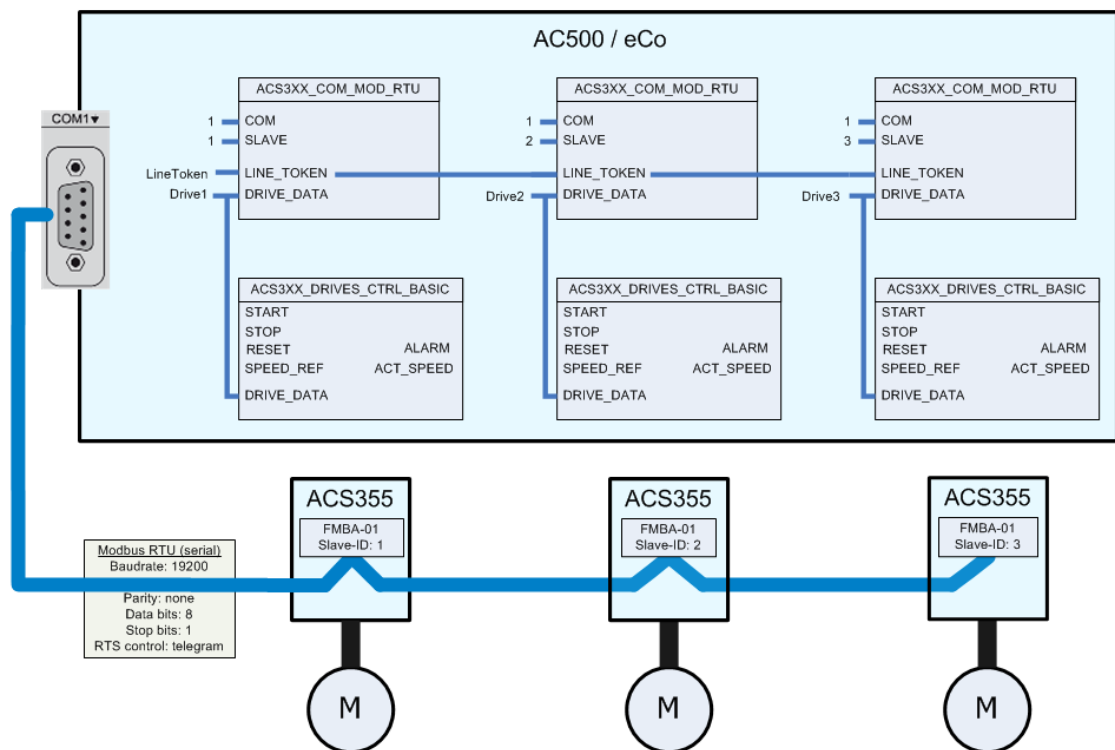
For the update to take effect, follow the steps below:

4. Click **Project > Rebuild all**.
5. Click **Online > Login**.



Adding another drive

The below instructions detail how to copy the program and the visualizations for another drive. The below image shows what the structure of the function blocks looks like when there are multiple drives.



18 Customization

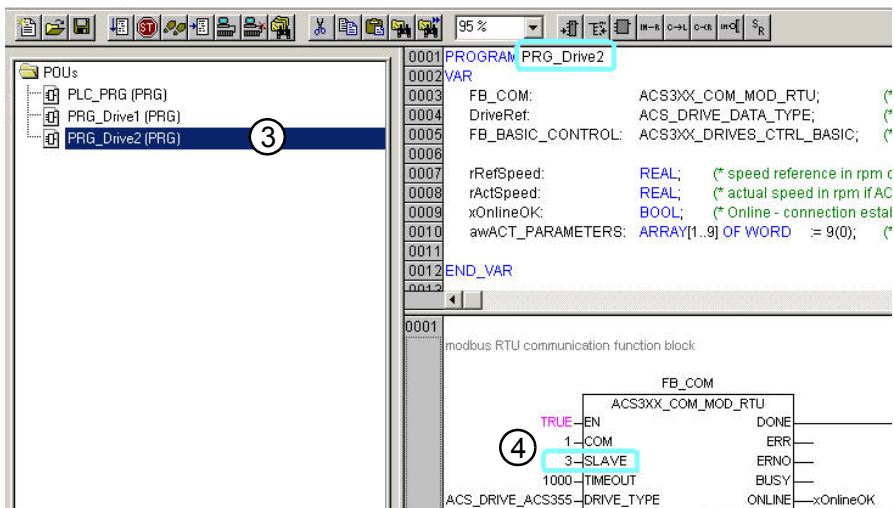
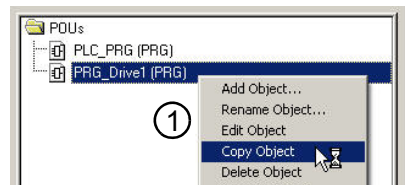
■ Copying the program for another drive

The instructions below outline how to copy the program of a drive in CoDeSys for a second drive.

1. In the left pane, in the **POUs** tab, right-click **PRG_Drive1 (PRG)** and click **Copy Object**.

A Copy Object dialog opens.

2. Change the name if necessary and click **OK**.

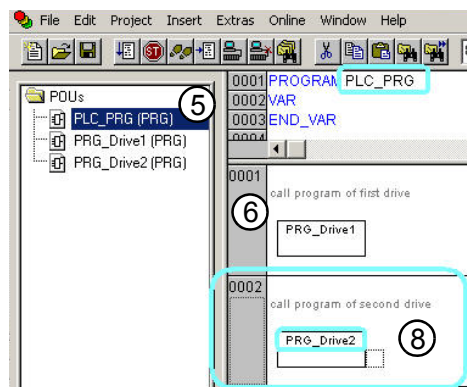


The new drive program appears in the POU's tab.

3. Double-click the new drive in the **POUs** tab.
4. In the **modbus RTU communication function block**, change the **SLAVE** number to match that of the new drive.
5. In the **POUs** tab, double-click **PLC_PRG (PRG)**.
6. Right-click the gray bar with **0001** on it (see the picture) and click **Copy**.
7. Right-click again and click **Paste**.

The copied instance of the program appears below the first one.

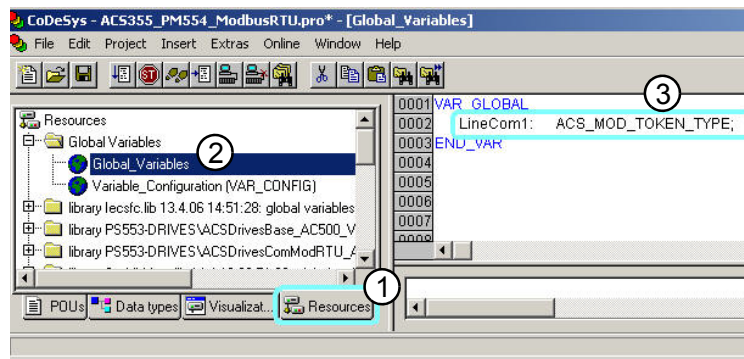
8. Double-click the name of the new block and change it to **PRG_Drive2**.
9. Click **Project > Rebuild all**.
10. Click **Online > Login**.
11. Click **Online > Create boot project**.



■ Information on LineCom1 as a global variable

The LineCom1 variable, which must be connected to all LINE_TOKEN inputs of all ACS3XX_COM_MOD_RTU function blocks in all programs, must be declared as a global variable. This is already done in the example program and can be seen below:

1. Open the **Resources** tab.
2. Select **Global_Variables**.
3. See declaration of LineCom1 of type ACS_MOD_TOKEN_TYPE.



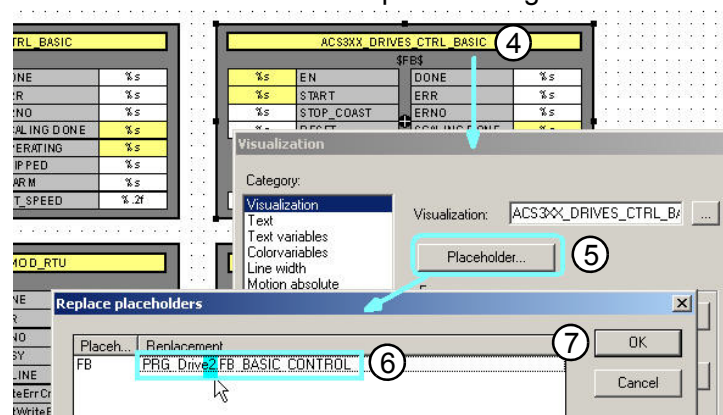
■ Creating a visualization for a copied drive program

You can copy the visual elements of a drive for another drive to enable monitoring each drive through the visualization.

1. Open the **Visualization** tab and double-click the PLC_VISU element to open the visualization.
2. Copy-paste the elements of the first drive (ie, right-click and **Copy**, then right-click and **Paste** each of them).
3. Arrange the copied elements, for example, next to the original ones.
4. Double-click a copied element.
5. With **Category: Visualization** selected, click **Placeholder....**

The Replace placeholders window opens.

6. In the Replacement tab, change the name, eg PRG Drive2 FB BASIC CONTROL.
7. Click **OK** and **OK** to accept the changes.



8. Repeat the previous steps for other copied elements.

Using visualizations

The visualization of a project can be used to monitor states in the function blocks, and errors and actual values. See the image below.

ACS3XX_DRIVES_CTRL_BASIC			
PRG_Drive1.FB_BASIC_CONTROL			
TRUE	EN	DONE	TRUE
TRUE	START	ERR	FALSE
FALSE	STOP_COAST	ERNO	0
FALSE	RESET	SCALING_DONE	TRUE
		OPERATING	TRUE
		TRIPPED	FALSE
		ALARM	FALSE
237.00	SPEED_REF	ACT_SPEED	702.00

State of the Function Block "FB_BASIC_CONTROL" in Programm "PRG_Drive1"

ACT_SPEED:
default rpm (ACS355 Par. 53.10 = 101)
0.1Hz if ACS355 Par. 53.10 = 103

ACS3XX_COM_MOD_RTU			
PRG_Drive1.FB_COM			
TRUE	EN	DONE	TRUE
1	COM	ERR	FALSE
1000	TIMEOUT	ERNO	0
2	SLAVE	BUSY	FALSE
4	DRIVE_TYPE	ONLINE	TRUE
		WriteErrCnt	0
		LastWriteErno	0
		ReadErrCnt	0
		LastReadErno	0
3	NVAR_READ	SW	4919
		ActValue1	702
		ActValue2	143
		ActValue3	0
		ActValue4	0
		ActValue5	0
		ActValue6	0
		ActValue7	0
		ActValue8	0

SPEED_REF:
if Control Mode = Scalar (default) then 0.1Hz
if Control Mode = Vector then rpm

internal counter of modbus write and read errors.
Reset by rising edge of EN.
Error code can be found in Online Help

actual values 1..8 are mapped in ACS3XX in Par 53.10 .. 53.17.

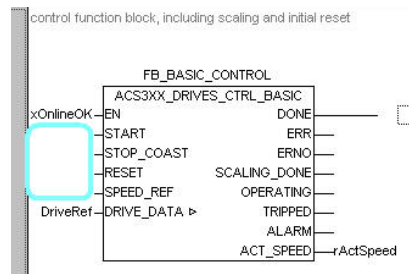
NVAR_READ input has to be set to number of read values (min. 3).

Controlling the drive with the visualization

To use the visualization to control the drive, disconnect all inputs of the function block to be controlled with the visualization in the program.

For example, to control START, STOP_COAST, RESET and SPEED_REF through the visualization, disconnect them from the FB_BASIC_CONTROL function block.

Once disconnected, you can set the input values directly in the visualization element. In this example, START, STOP_COAST, RESET and SPEED_REF can be set through the visualization.



ACS3XX_DRIVES_CTRL_BASIC			
PRG_Drive1.FB_BASIC_CONTROL			
TRUE	EN	DONE	TRUE
FALSE	START	ERR	FALSE
FALSE	STOP_COAST	ERNO	0
FALSE	RESET	SCALING_DONE	TRUE
		OPERATING	FALSE
		TRIPPED	FALSE
		ALARM	FALSE
975.00	SPEED_REF	ACT_SPEED	0.00

Configuring parameters to be read from the drive

The ACS3XX_COM_MOD_RTU function block can read up to 8 parameters from the ACS3xx drive.

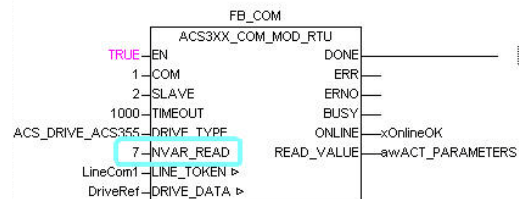
The ACS3xx drive parameters 53.10...53.17 determine which parameters are read from the drive. Give the values in the format XXYY, where XX is the parameter group and YY is the parameter index of the parameter to be read. For example, the value 104 corresponds to parameter 01.04 CURRENT.

Set parameter 53.10 to either 101 (calculated motor speed) or 103 (calculated drive output frequency). You can freely choose which parameters to map to parameters 53.11...53.17.

For instructions on how to set parameter values, see *ACS355 user's manual* (3AUA0000066143 [English]).

If more than three parameters are configured to be read, change the value of NVAR_READ in the ACS3XX_COM_MOD_RTU function block as instructed below.

1. In CoDeSys, go offline by clicking **Online > Logout**.
2. In the ACS3XX_COM_MOD_RTU function block, change the value of NVAR_READ to correspond to the number of parameters read + 1 (the number includes the status word).



3. Go back online by clicking **Online > Login**.

An Online Change dialog opens.

4. Click **Yes**.

The visualization is updated to reflect the changes made.

5. To keep the changes for the next reboot, click **Online > Create boot project**.

■ Configuration example

In this example, six parameters are configured to be read from an ACS355 drive by the function block. The following values are given to parameters 53.10...53.16:

Parameter set in the drive	Value	Parameter read from the drive	Function
53.10	101	01.01 SPEED & DIR	Calculated motor speed in rpm. A negative value indicates reverse direction.
53.11	104	01.04 CURRENT	Measured motor current (1 = 0.1 A).
53.12	305	03.05 FAULT WORD 1	A 16-bit data word that indicates faults in the drive.
53.13	110	01.10 DRIVE TEMP	Measured IGBT temperature in (1 = 0.1 °C).
53.14	114	01.14 RUN TIME (R)	Elapsed drive running time in hours.
53.15	105	01.05 TORQUE	Ratio of calculated motor torque to the motor nominal torque (1 = 0.1% of the nominal torque).

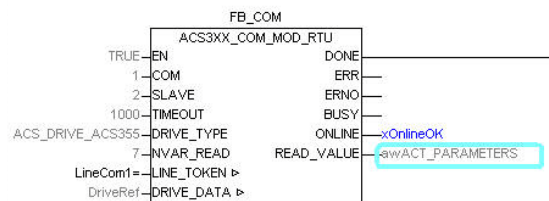
22 Customization

The value of the NVAR_READ input is changed accordingly. Because there are six parameters to be read (plus one for the status word), NVAR_READ is changed to 7 as instructed above. Once the program is back online, the visualization changes to show the values of the parameters read from the drive:

ACS3XX_COM_MOD_RTU			
PRG_Drive1_FB_COM			
TRUE	EN	DONE	FALSE
1	COM	ERR	FALSE
1000	TIMEOUT	ERNO	0
2	SLAVE	BUSY	FALSE
4	DRIVE_TYPE	ONLINE	TRUE
		WriteErrCnt	0
		LastWriteErno	0
		ReadErrCnt	0
		LastReadErno	0
7	NVAR_READ	SW	4919
		ActValue1	702
		ActValue2	7
		ActValue3	0
		ActValue4	368
		ActValue5	3
		ActValue6	102
		ActValue7	0
		ActValue8	0

Using the read parameters in the program

The parameters read from the drive appear as variables named awACT_PARAMETERS[1...9] in CoDeSys. The first index in the array corresponds to the status word, and indexes 2...9 correspond to the parameters mapped to be read. For example, if parameter 53.13 is set to 110 (01.10 DRIVE TEMP), awACT_PARAMETERS[5] gets the value of the parameter.

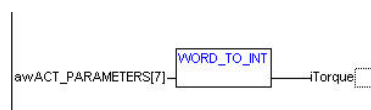


```

0001 FB_COM
0002 DriveRef
0003 FB_BASIC_CONTROL
0004 rRefSpeed = 237
0005 rActSpeed = 702
0006 xOnlineOK = TRUE
0007 awACT_PARAMETERS
0008 awACT_PARAMETERS[1] = 4919
0009 awACT_PARAMETERS[2] = 702
0010 awACT_PARAMETERS[3] = 7
0011 awACT_PARAMETERS[4] = 0
0012 awACT_PARAMETERS[5] = 368
0013 awACT_PARAMETERS[6] = 3
0014 awACT_PARAMETERS[7] = 102
0015 awACT_PARAMETERS[8] = 0
0016 awACT_PARAMETERS[9] = 0
0017
  
```

Word-to-integer conversion for variables with negative values

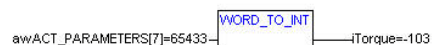
The data type (word) of the READ_VALUE output values in the ACS3XX_COM_MOD_RTU function block permits only non-negative values, but it is possible to use a word-to-integer conversion to enable also negative values. In the following example, awACT_PARAMETERS[7] is converted to a new variable, iTorque which represents the torque and direction of motor rotation.



Declare the iTorque variable as integer:

iTorque: INT;

The value is then converted into an integer as shown below. The negative value represents reverse direction in this example.



4

Function blocks and program structure

What this chapter contains

This chapter presents an overview of the function blocks of the ACS Drives Libraries.

Function block programming overview

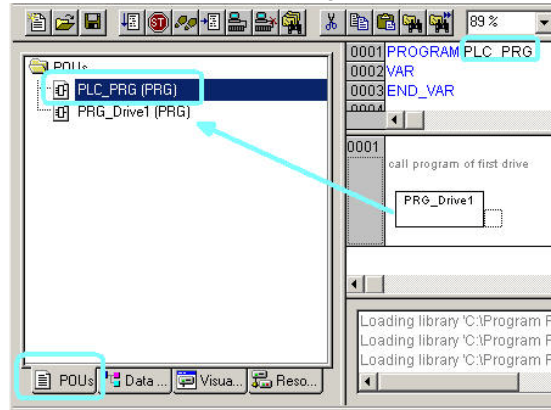
The main advantage of using function blocks is that it is easy to control drives and there is no need to master the details of the ABB Drives profile.

Function block programming also allows for a modular design and customization for the requirements of the application in question. The ACS Drives Base Library in the starter kit features several types of control blocks, for both basic and advanced applications.

Program structure

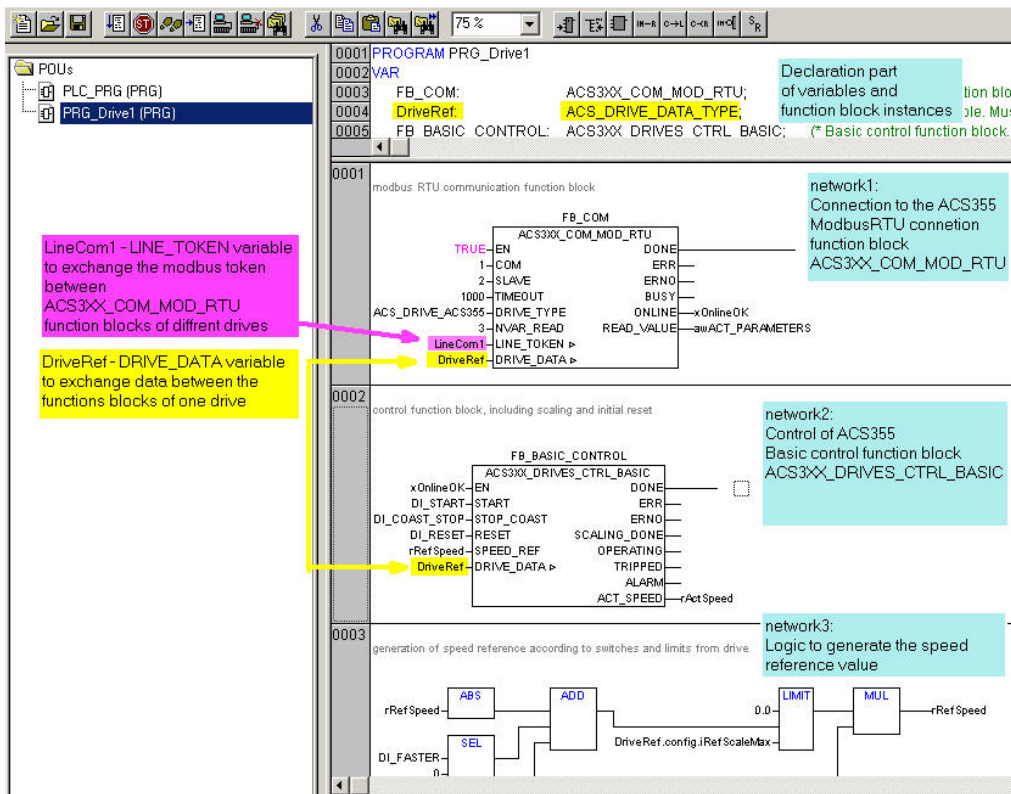
Main program

In the example project, the main program is named PLC_PRG. The main program includes a call to PRG_Drive1 (see below). If there are multiple drives, add a program for each additional drive (see [Adding another drive](#) on page 17).



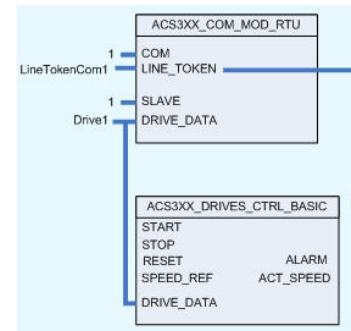
PRG_Drive1 program

PRG_Drive1 contains the function blocks for the drive. The image below shows the main components of PRG_Drive1 (FB_COM, FB_BASIC_CONTROL and speed reference) and their functions.



■ DriveRef

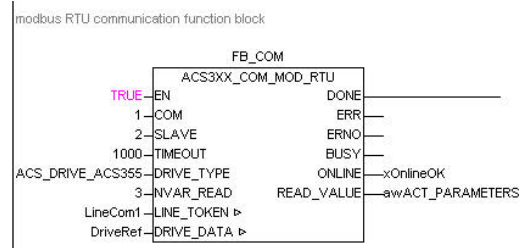
The DriveRef (DRIVE_DATA) variable must be connected to all function blocks of the drive to exchange data.



Main components of the example program

■ Modbus RTU communication function block (FB_COM)

FB_COM is a communication function block for Modbus RTU. It controls the communication to the drive and cyclically reads the status word, actual speed, and up to seven more parameters from the drive. When there are changes in the control word or the speed reference values, the function block writes the new values in the drive.



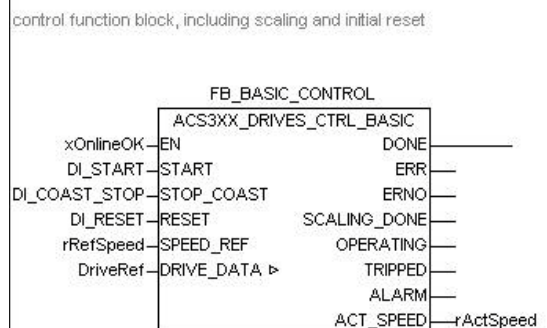
Refer to the online help for a detailed description of the function block and its inputs and outputs. For additional instructions, see the below sections in this manual:

- You can set the slave address as described in [Configuring the slave address of the drive](#) on page 17.
- If more than one drive is used, the LineCom1 variable in the LINE_TOKEN input must be declared as global. See [Information on LineCom1 as a global variable](#) on page 18.
- For a description of the DRIVE_DATA variable DriveRef, see [DriveRef](#) on page 25.
- If more parameters are to be read from the drive, adjust NVAR_READ and set the parameters (53.10...53.17) in the drive. See [Configuring parameters to be read from the drive](#) on page 21.

■ Basic control block (FB_BASIC CONTROL)

The FB_BASIC_CONTROL function block is used for basic control of the drive, including speed scaling of the actual speed and reference value to rpm or 0.1 Hz.

The FB_BASIC CONTROL function block is the default control block in the example project. It is interchangeable with the function blocks presented in [Other control blocks](#) below. A drive may have one active control block.



See the online help for detailed information on the inputs and outputs of the function block. See [Using the online help in CoDeSys](#) on page 15 for information on the use of the online help.

The function block functions only with the ABB Drives profile.

26 Function blocks and program structure

■ Speed reference generation

In the example program, speed reference is generated according to the dip switches (DI_FASTER, DI_SLOWER and DI_REVERS) and the maximum reference limit.

The maximum reference limit is read from the drive parameter 11.05 (REF1 MAX) at the first start of the ACS3XX_DRIVES_CTRL_BASIC function block and is stored in config.iRefScaleMax, a sub-element of the DriveRef variable.

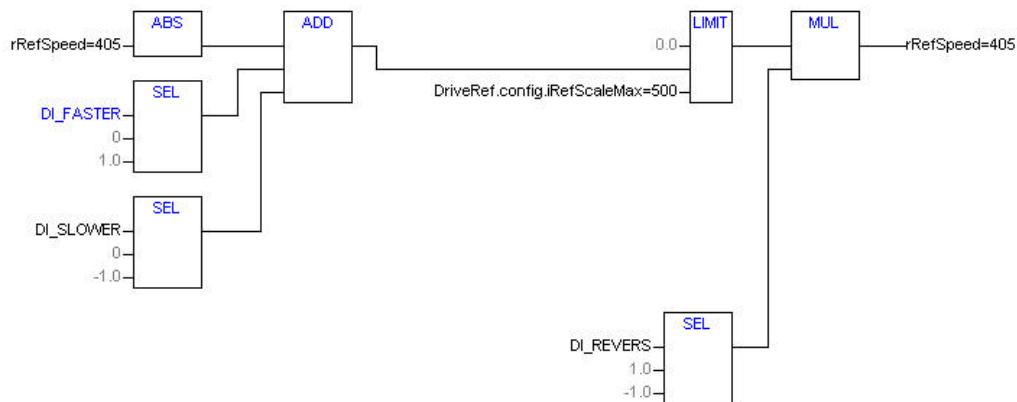
The limit depends on the selected motor control mode. There are two possible motor control modes, detailed below. Drive parameter 99.04 Motor Control Mode determines the mode used.

- **Scalar mode:** The default mode is the scalar mode. The default value for the drive parameter REF1 MAX is 500 (which corresponds to 50 Hz).
- **Vector mode:** In the vector mode, the drive parameter REF1 MAX is set to the maximum speed given in rpm, eg 1500.

If you change the motor control mode, the scaling parameter has to be read again by resetting EN of ACS3XX_DRIVES_CTRL_BASIC. This can be done by powering off and on again the 24V DC of the AC500.

DI_FASTER, DI_SLOWER and DI_REVERS are used in the example project as shown in the below image. See the quick installation guide for the dip switch settings in the ACS355 and AC500-eCo starter kit.

generation of speed reference according to switches and limits from drive

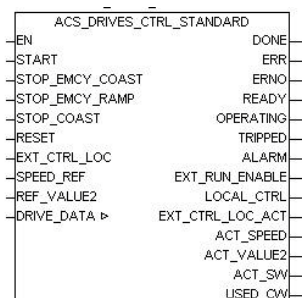


Other control blocks

You can use the following two control blocks as alternatives for the basic control block.

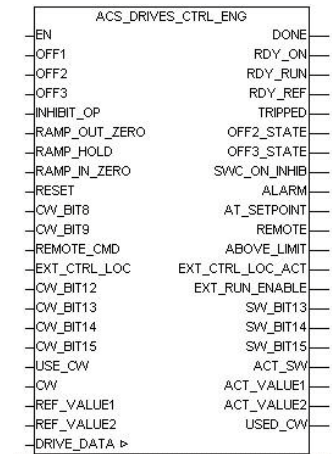
■ ACS_CTRL_ABB_DRV_PROFILE

If advanced control is needed, the ACS_DRIVES_CTRL_STANDARD function block can be used instead of the FB_BASIC_CONTROL function block. Refer to the online help for a detailed description of the function block.



■ ACS_DRIVES_CTRL_ENG

For system applications, the engineering block ACS_DRIVES_CTRL_ENG can be used. Refer to the online help for a detailed description of the function block.



28 *Function blocks and program structure*



Contact us

ABB STOTZ-KONTAKT GmbH

P.O. Box 101680
69006 Heidelberg, Germany
Telephone +49 (0) 6221 701-0
Fax +49 (0) 6221 701-240
E-Mail plc.support@de.abb.com

www.abb.com/plc

ABB Oy

Drives
P.O. Box 184
FI-00381 HELSINKI
FINLAND
Telephone +358 10 22 11
Fax +358 10 22 22681
www.abb.com/drives

Note:

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 © 2011 ABB

All rights reserved

Order number 2CDC 125 152 M0201