Application guide
ACS355 and AC500-eCo
List of related manuals

<table>
<thead>
<tr>
<th>Drive and PLC hardware manuals and guides</th>
<th>Code (English)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACS355 user’s manual</td>
<td>3AU0000066143</td>
</tr>
<tr>
<td>AC500-eCo and ACS355 quick installation guide</td>
<td>2CDC125145M0201</td>
</tr>
<tr>
<td>System description AC500</td>
<td>2CDC125015M0201</td>
</tr>
<tr>
<td>PM554 and PM564 installation instructions</td>
<td>2CDC125122M6801</td>
</tr>
<tr>
<td>AC500 online help</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option manuals and guides</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FMBA-01 Modbus adapter module user’s manual</td>
<td>3AFE68586704</td>
</tr>
</tbody>
</table>

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.
# 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
  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

## 4. Function blocks and program structure
What this chapter contains ......................................... 23
Function block programming overview ....................... 23
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program structure</td>
<td>24</td>
</tr>
<tr>
<td>Main program</td>
<td>24</td>
</tr>
<tr>
<td>PRG_Drive1 program</td>
<td>24</td>
</tr>
<tr>
<td>DriveRef</td>
<td>25</td>
</tr>
<tr>
<td>Main components of the example program</td>
<td>25</td>
</tr>
<tr>
<td>Modbus RTU communication function block (FB_COM)</td>
<td>25</td>
</tr>
<tr>
<td>Basic control block (FB_BASIC CONTROL)</td>
<td>25</td>
</tr>
<tr>
<td>Speed reference generation</td>
<td>26</td>
</tr>
<tr>
<td>Other control blocks</td>
<td>26</td>
</tr>
<tr>
<td>ACS_CTRL_ABB_DRV_PROFILE</td>
<td>26</td>
</tr>
<tr>
<td>ACS_DRIVES_CTRL_ENG</td>
<td>27</td>
</tr>
</tbody>
</table>
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.
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>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.</td>
<td>LOC 'MAIN MENU' PAR</td>
</tr>
<tr>
<td>2.</td>
<td>Go to the Parameters mode by selecting PARAMETERS on the menu with keys and , and pressing</td>
<td>LOC 'PAR GROUPS' 999</td>
</tr>
<tr>
<td>3.</td>
<td>Select the appropriate parameter group with keys and . Press</td>
<td>LOC 'PARAMETERS' 999</td>
</tr>
<tr>
<td>4.</td>
<td>Select the appropriate parameter with keys and . The current value of the parameter is shown below the selected parameter. Press</td>
<td>LOC 'PARAMETERS' 999</td>
</tr>
<tr>
<td>5.</td>
<td>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.</td>
<td>LOC 'PARAMETERS' 999</td>
</tr>
<tr>
<td>6.</td>
<td>• To save the new value, press . • To cancel the new value and keep the original, press</td>
<td>LOC 'PARAMETERS' 999</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>EXT1 COMMANDS</td>
<td>COMM (10)</td>
</tr>
<tr>
<td>1102</td>
<td>EXT1/EXT2 SEL</td>
<td>COMM (8)</td>
</tr>
<tr>
<td>1103</td>
<td>REF1 SEL</td>
<td>COMM (8)</td>
</tr>
<tr>
<td>1604</td>
<td>FAULT RESET SEL</td>
<td>COMM (8)</td>
</tr>
</tbody>
</table>
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.

- 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.
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.
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTS control</td>
<td>telegram</td>
</tr>
<tr>
<td>Baudrate</td>
<td>(same as in the drive)</td>
</tr>
<tr>
<td>Parity</td>
<td>(same as in the drive)</td>
</tr>
<tr>
<td>Data bits</td>
<td>(same as in the drive)</td>
</tr>
<tr>
<td>Stop bits</td>
<td>(same as in the drive)</td>
</tr>
<tr>
<td>Operation mode</td>
<td>Master</td>
</tr>
<tr>
<td>Address</td>
<td>0</td>
</tr>
</tbody>
</table>

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.
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 **POUs** 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.

### 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.
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:

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

### 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.

#### 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:

```plaintext
iTorque: INT;
```

The value is then converted into an integer as shown below. The negative value represents reverse direction in this example.
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.
Function blocks and program structure 25

- **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.
**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.

**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.
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