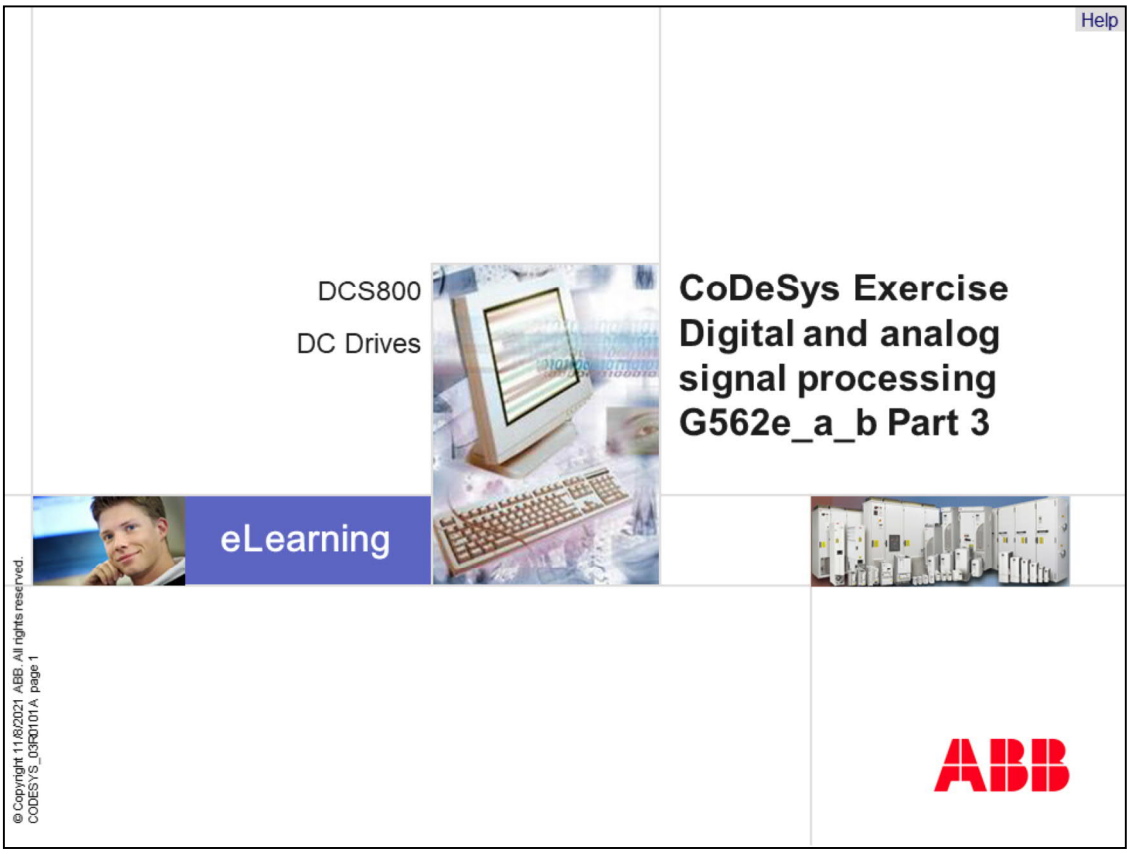


Welcome to the CoDeSys training module for the DCS800, ABB DC Drives.

If you need help navigating this module, please click the Help button in the top right-hand corner. To view the presenter notes as text, please click the Notes button in the bottom right corner.

Help

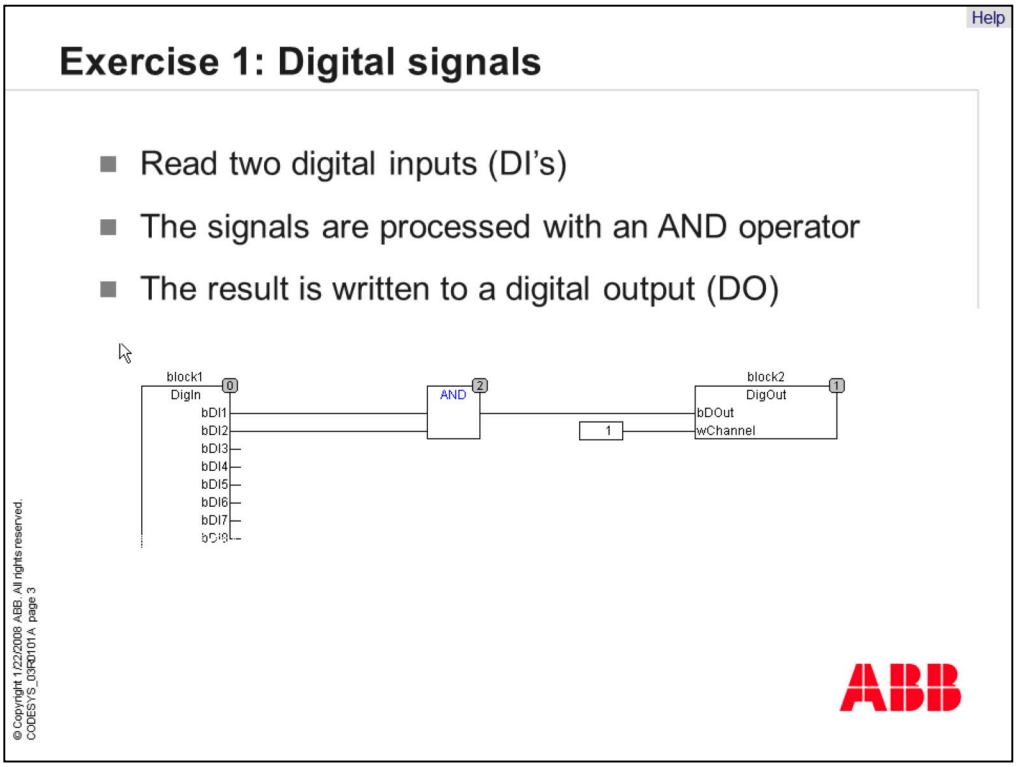
Objectives

After completing this module, you will be able to

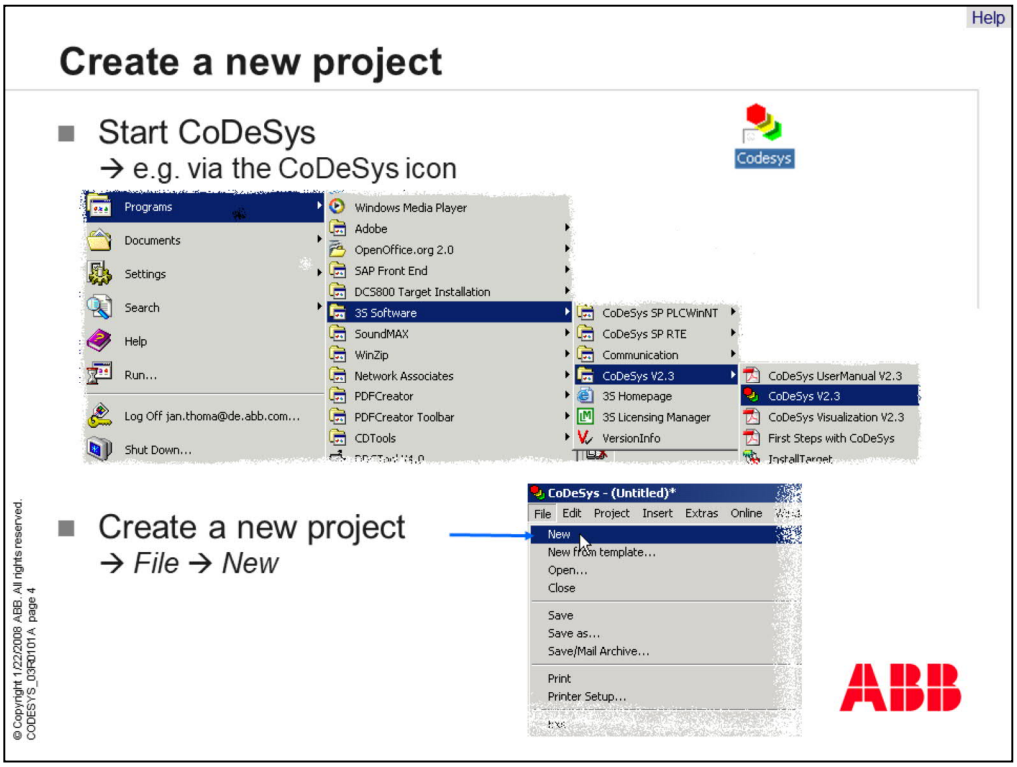
- Create a small program
- Document an application
- Set the task configuration
- Download a program into the drive

ABB

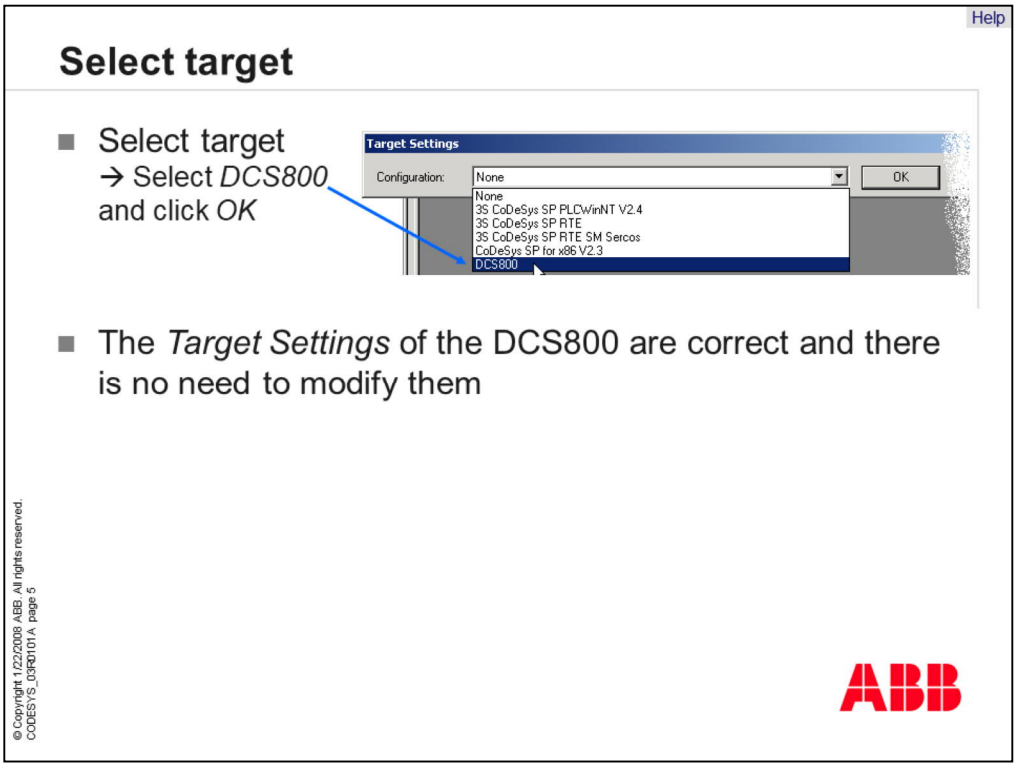
- After completing this module, you will be able to
- understand the basics of the CoDeSys programming tool,
- create small applications,
- document your applications for better understanding,
- download a program into the drive,
- evaluate the risks for each application.



Let's start with the first example. It is a small program with three blocks which should handle the digital inputs and outputs of the drive. Two of the function blocks are specifically designed for DCS800. The first one reads the digital inputs. It is called "DigIn". By using the "DigOut" block you can set the digital outputs of the drive. The blocks are connected with an "AND operator" which can be found in the list with all IEC operators.

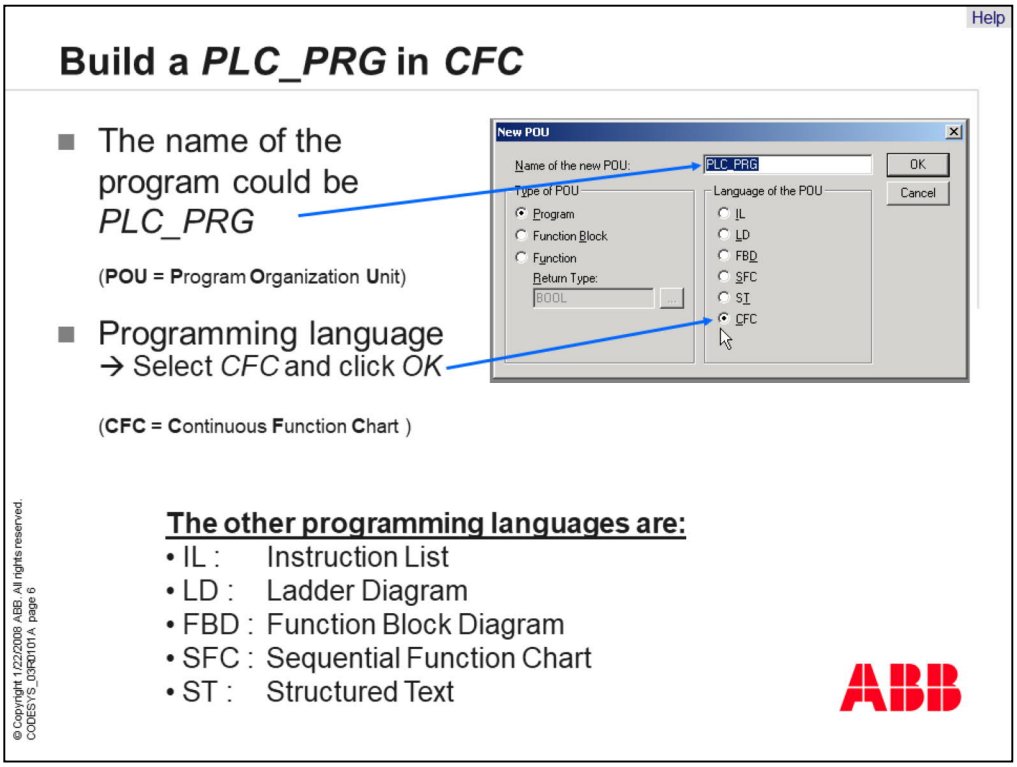


You can start CoDeSys by using a shortcut on the desktop or by finding CoDeSys in the start-menu. After CoDeSys has started up the last program edited will open automatically. To create a new application, click on “New” in the „File“ menu.

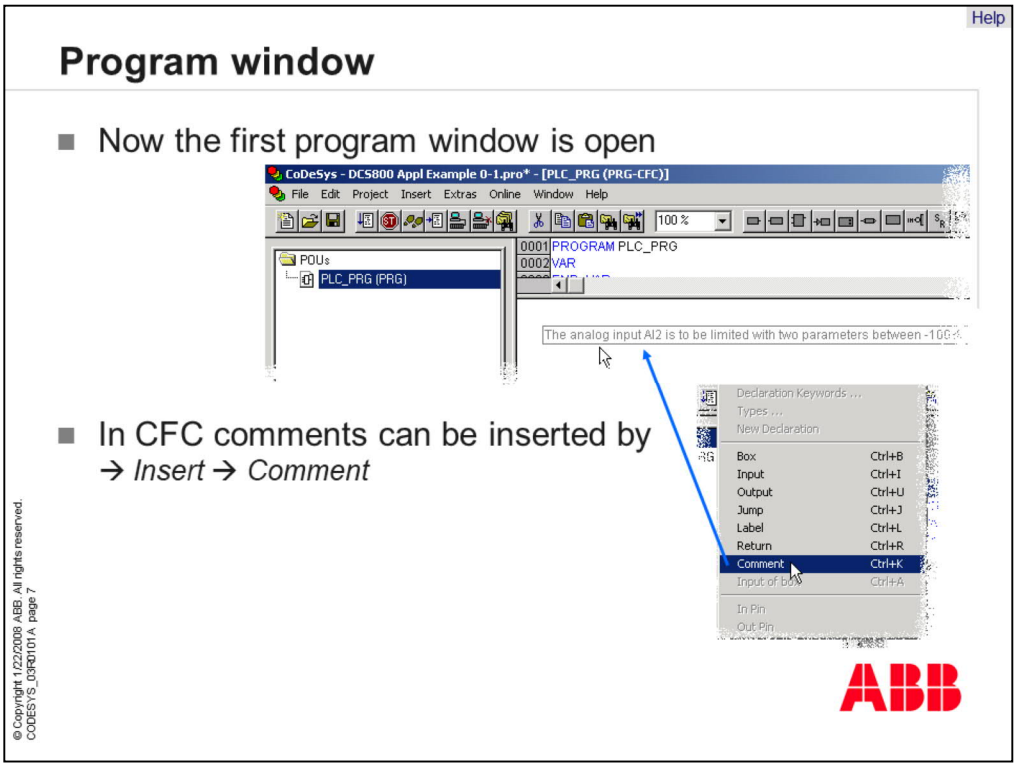


Select the DCS800 as target to get the correct settings between the drive processor and the software tool. If other PLCs are available, you can see them in the pull-down-menu, for example the “AC500”, and so on. The target file includes information about the communication channel and memory functions of the DCS800. For the DCS800 a serial communication between the PC and the drive is required.

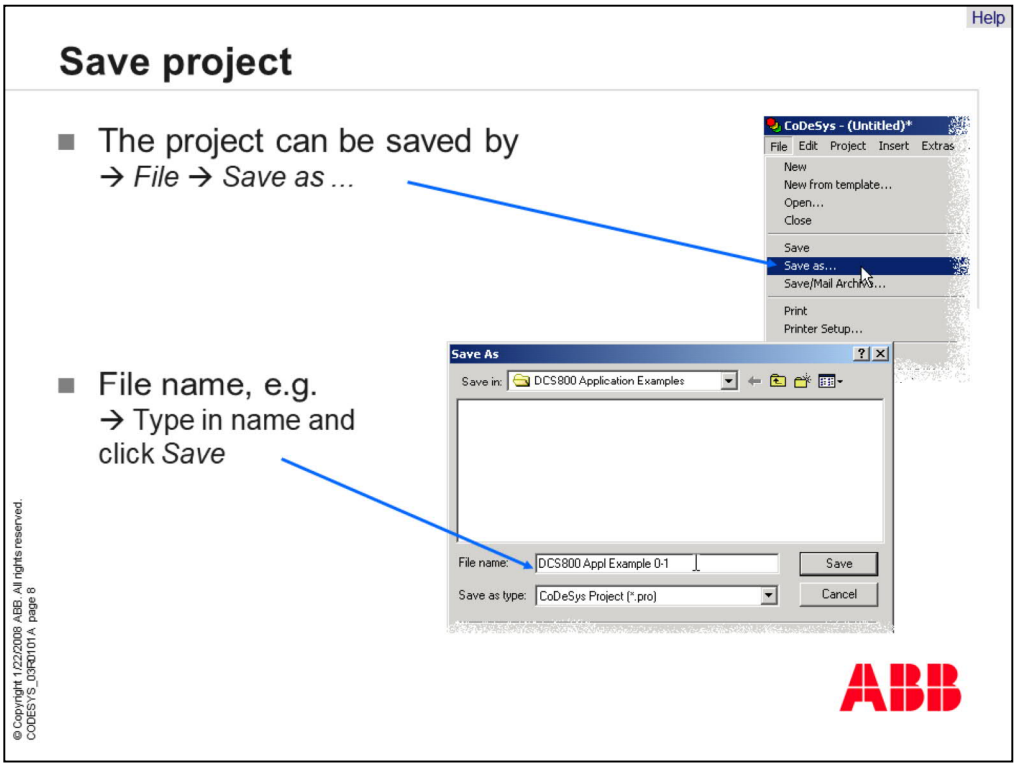
Note, programs can be simulated by using the target setting “None”. The special function blocks for the DCS800 cannot be simulated with the system.



CoDeSys includes many programming languages. The „Instruction List“ is a text-based language. Programming in the language „Ladder Diagram“ is like a circuit schematic with symbols. It is often used in America. „Function Block Diagram“ is a graphic-based language with different function blocks. „Sequential Function Chart“ is a language which splits the program process into steps. „Structured Text“ is a text-based language like C-Code or Pascal. „Continuous Function Chart“ is a free programmable graphic-based language with function blocks.

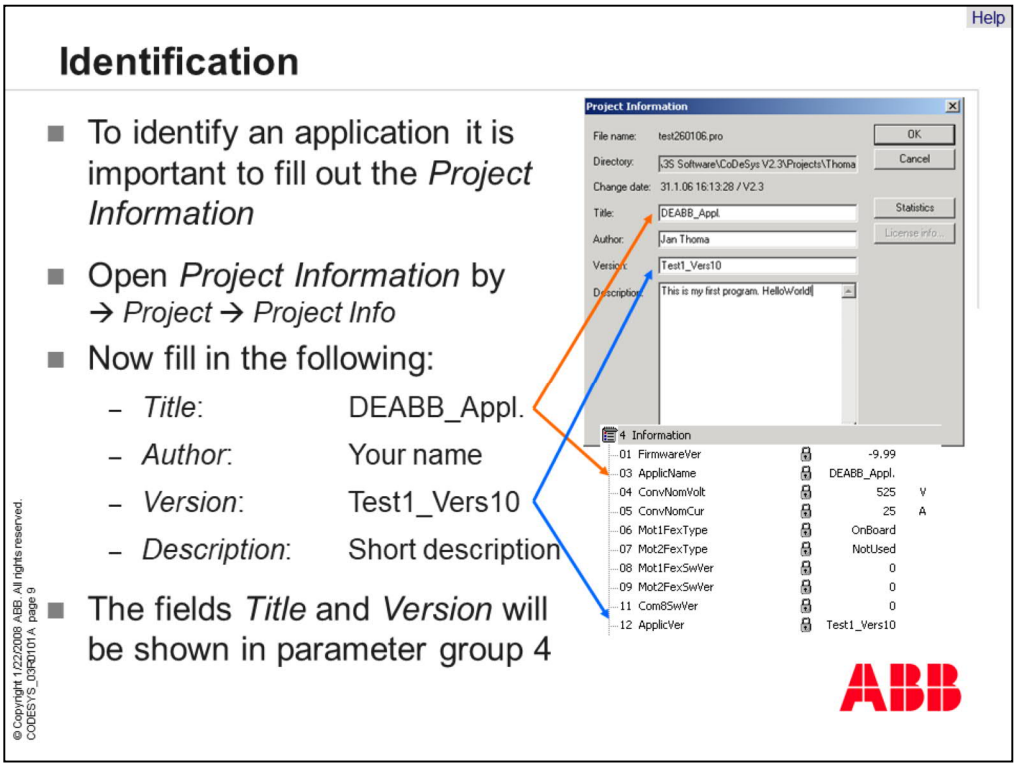


Good programming style is signalized by perfect documentation. In CoDeSys it is possible to add comments for each step in the program. A click with the right mouse button in the “Editor Window” opens the context-menu from which the item „Comment“ can open the comment dialog box. In other programming languages this functionality is also available and should be used to facilitate understanding of the program for others. The last step of project development is to save the project.



The next step is to save the project in a program file with the extension “*.pro”. Choose “Save as…” in the “File” menu, type in a file name for the program and click “Save”. Now the application is saved.

It is recommended to save each project in a separate folder on your PC together with a description of the functionality for each file. It’s also possible to print the complete project.

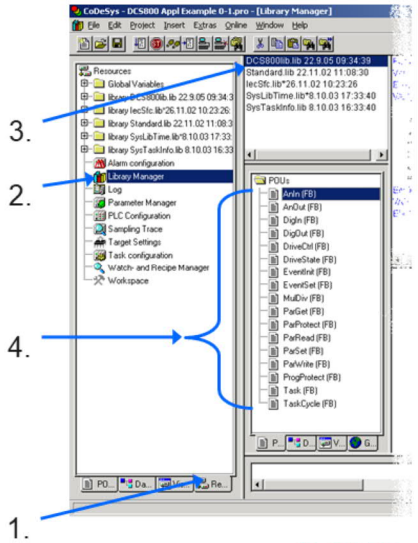


Providing a title and specifying a version of the application will allow for easier identification of the program. The project information will be shown in parameters of group 4. Parameter 4.03 shows the title field and 4.12 the version field. If no application is active, parameter 4.03 will show “No Application”.

Help

DCS800 library

1. Select tab *Resources*
2. Double click on *Library Manager*
3. Select *DCS800lib.lib*
4. Existing function blocks of the DCS800 library



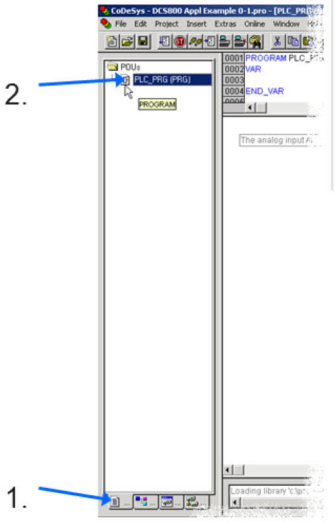
All DCS800 interface function blocks can be found in the DCS800 library. It is an external library which only works together with the DCS800 converter. Please note that the actual firmware-version of the drive should fit to the version of DCS800 library!

Inside the library you can find function blocks for reading and writing analog and digital values, event handling and arithmetic calculations. With CoDeSys it is also possible to read and write parameters with special DCS800 function blocks.

Help

Change back to program window

1. Select tab *POU*
2. Double click on *PLC_PRG* to open the program window

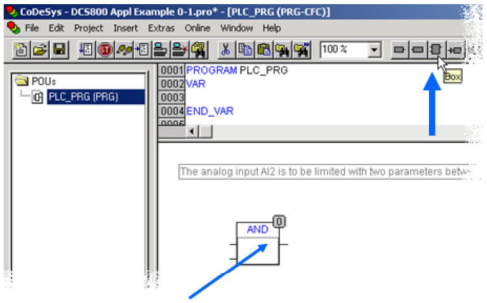


In CoDeSys the abbreviation “POU” stands for “program organization unit”. All programs, functions and function blocks can be found in the root directory. By clicking on a program in the directory you will open that program’s window. The default program name is PLC_PRG. It is possible to change this name because for the DCS800 it is not necessary to have a main program.

Help

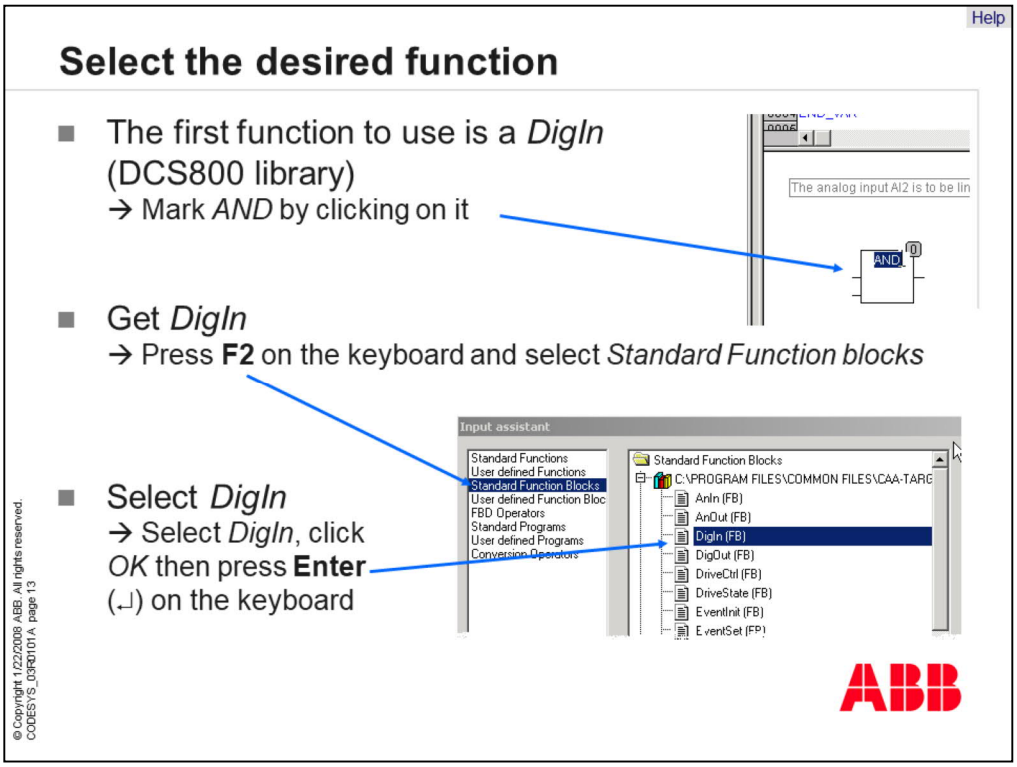
Insert function blocks

- Select the *Box* icon

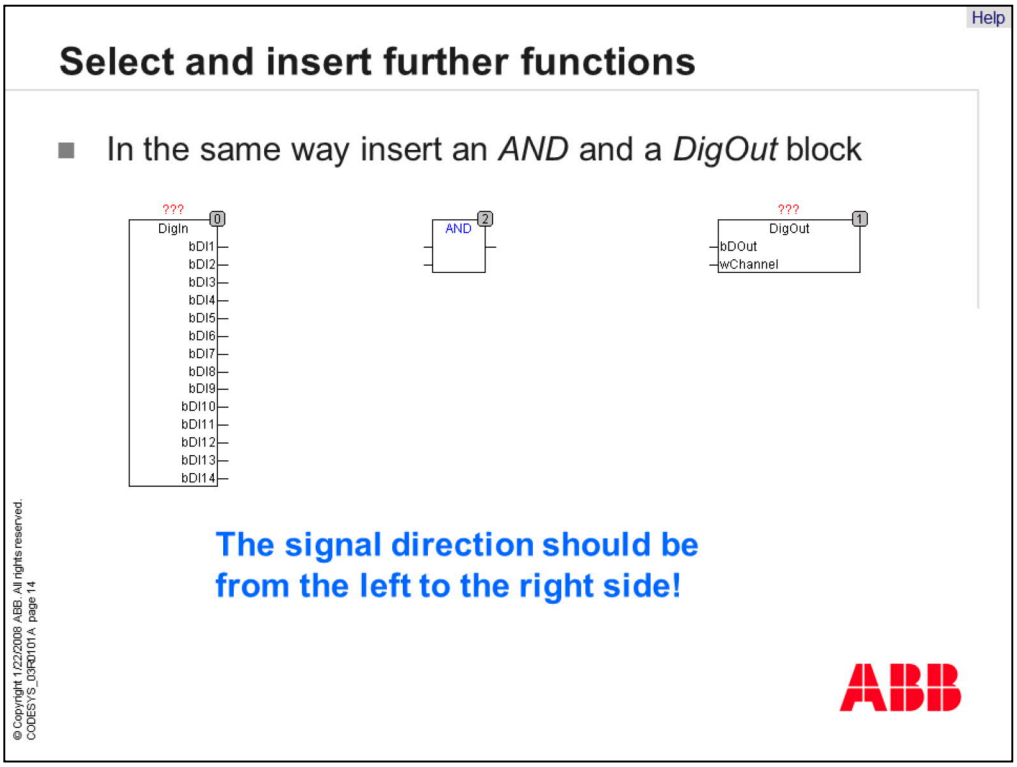


- Drag the *Box* into the Editor window. The inserted box is always an *AND* - operator

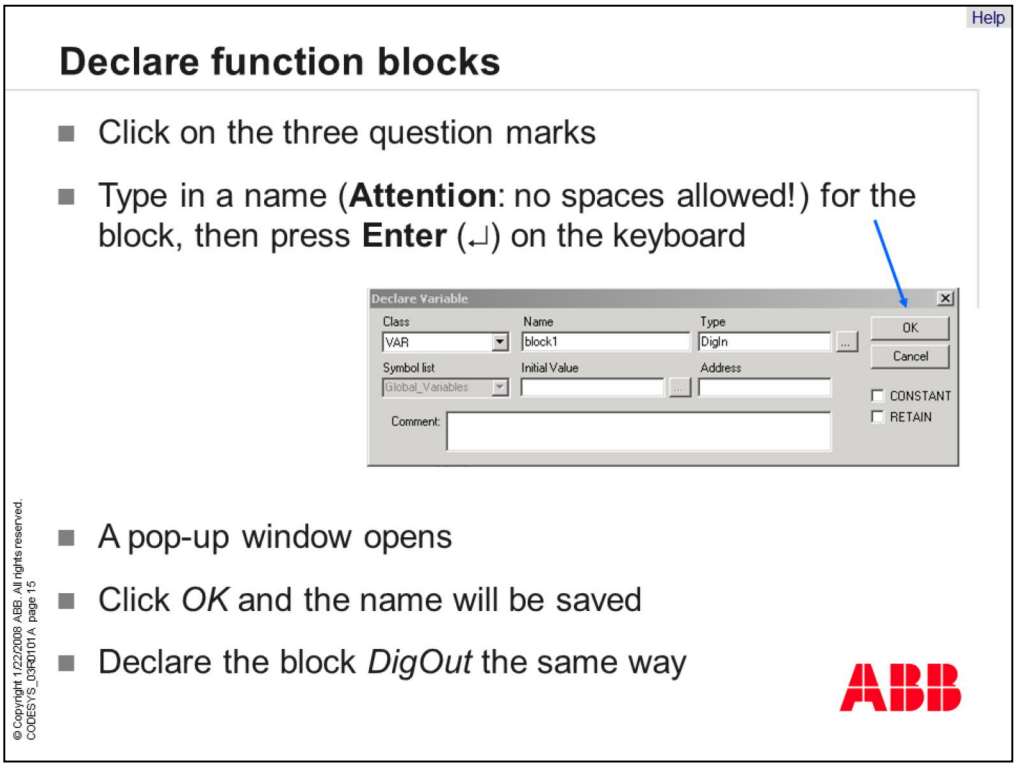
To insert a function block, please select the icon “box” in the menu. Now the box can be placed in the editor window. On the right side of each box, you will find a number. This number gives you information about the execution order of the function blocks. In a network of many function blocks it is necessary to check the correct execution order. An example is a multiplication of two variables. A multiplication should not be performed until both variables are updated. Otherwise, the calculation would be inaccurate.



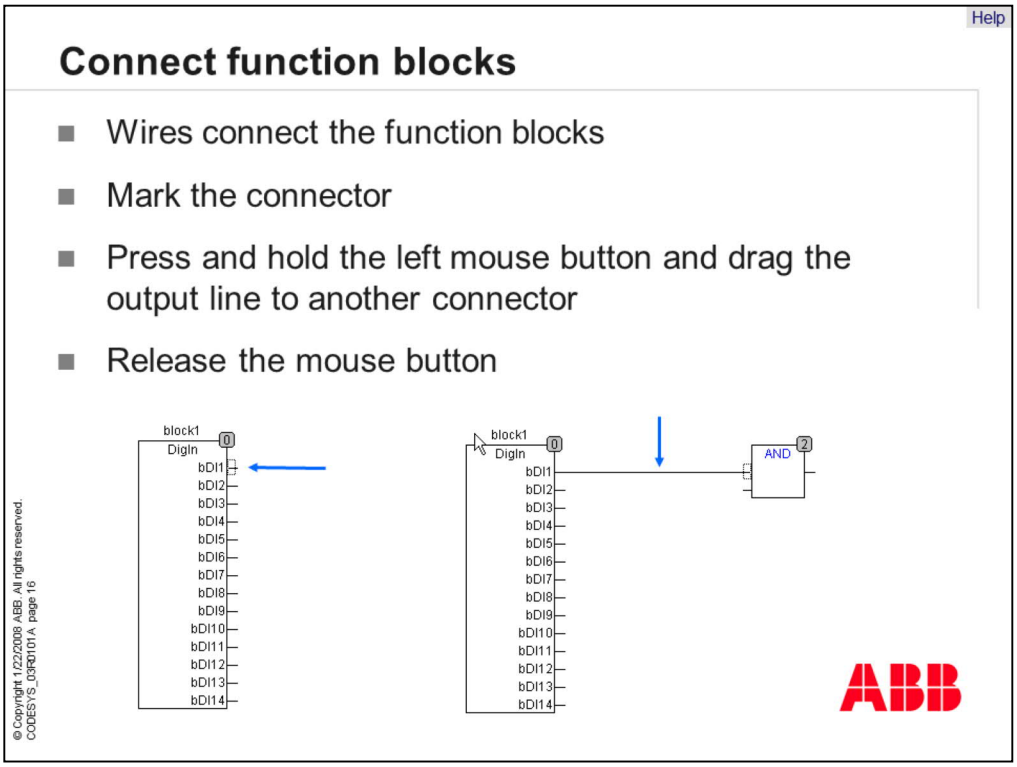
After a new block is inserted, the “block type” must be changed. To do this, select the type and press the F2 button. A new window with a list of all available operators and libraries with the several function blocks will open. In the menu “Standard Function Blocks” you can find the available function blocks of the several libraries. The menu “FBD Operators” includes all IEC functions like AND, OR, MUL and so on.



Two further function blocks are needed for this application program. The next step is to place an ”AND” operator and a “DigOut” function block to the correct position in the editor window. In the programming language “continuous function chart” boxes can be adjusted freely in the editor window. Note that the numbers on the upper right side of the boxes show the execution order of the blocks. Normally the execution order should be from left to right according to the signal direction.



Each function block should be named for identification purposes. This name is called “Instance”. Click on the question marks on the function block and specify a name. A new window opens and shows the name of the block and the used type. By clicking “OK” the settings will be saved, and the name will be shown in the declaration window. These steps are required to use any function block. Operators and functions don’t get a declaration.

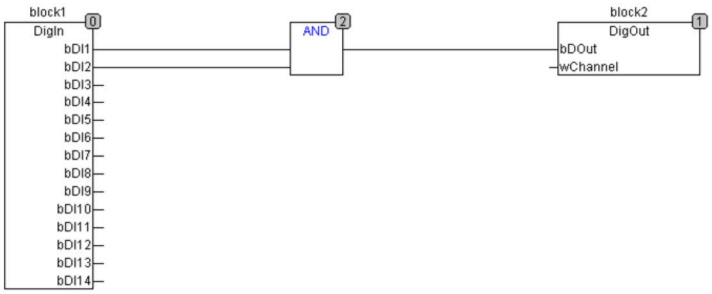


In programming language “CFC” the blocks have to be connected with wires. Click on an input or output and draw a line to another connecting point. It is not possible to connect outputs with each other. Note, in other programming languages a connection of inputs and outputs are drawn automatically.

Help

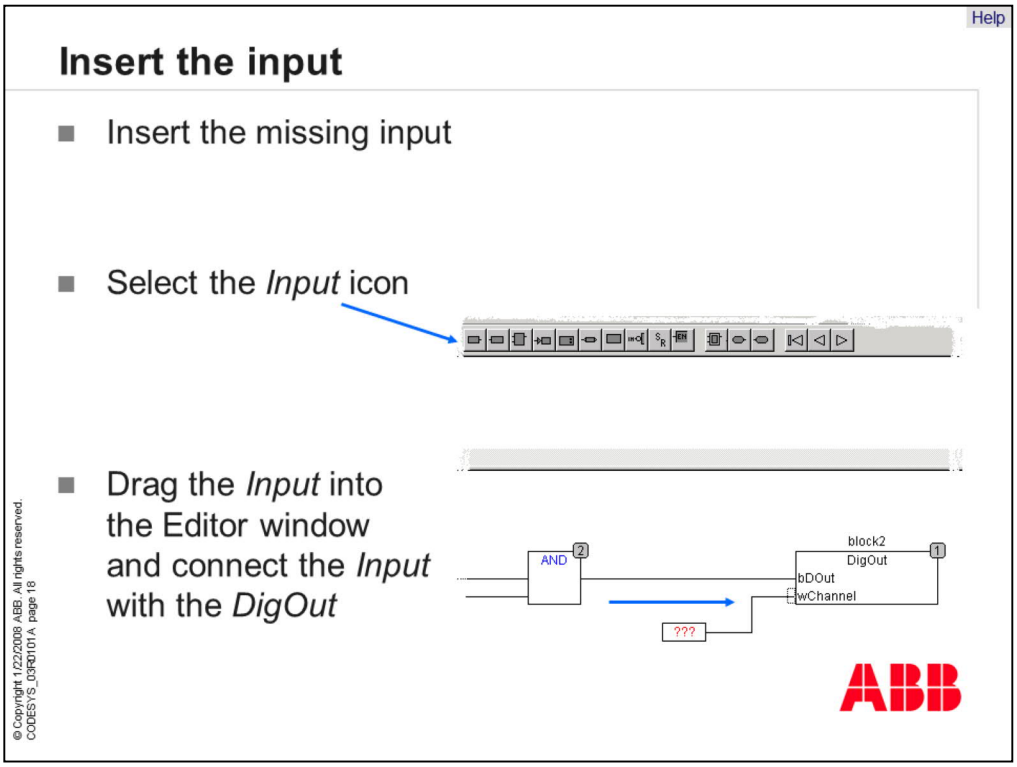
Connect function blocks

- Use the same method to make up all other connections between the function blocks



Connect all necessary function blocks in a similar fashion. The end result should look like the graphic with the three boxes.

In text-based programming languages like “Structured Text” or “Instruction List” connections are done with allocations. These connections are invisible but represented by variables!

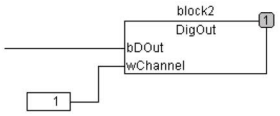


Now the boxes are connected with each other, but a constant value is needed for the input “Channel” of function block “DigOut”. The next step is to place an input box. Drag the input box into the editor window and connect it with the connector of function block “DigOut”.

Help

Declare the input value

- Click on the three question marks
- Type in a number for the *Input*, then press **Enter** (↵) on the keyboard

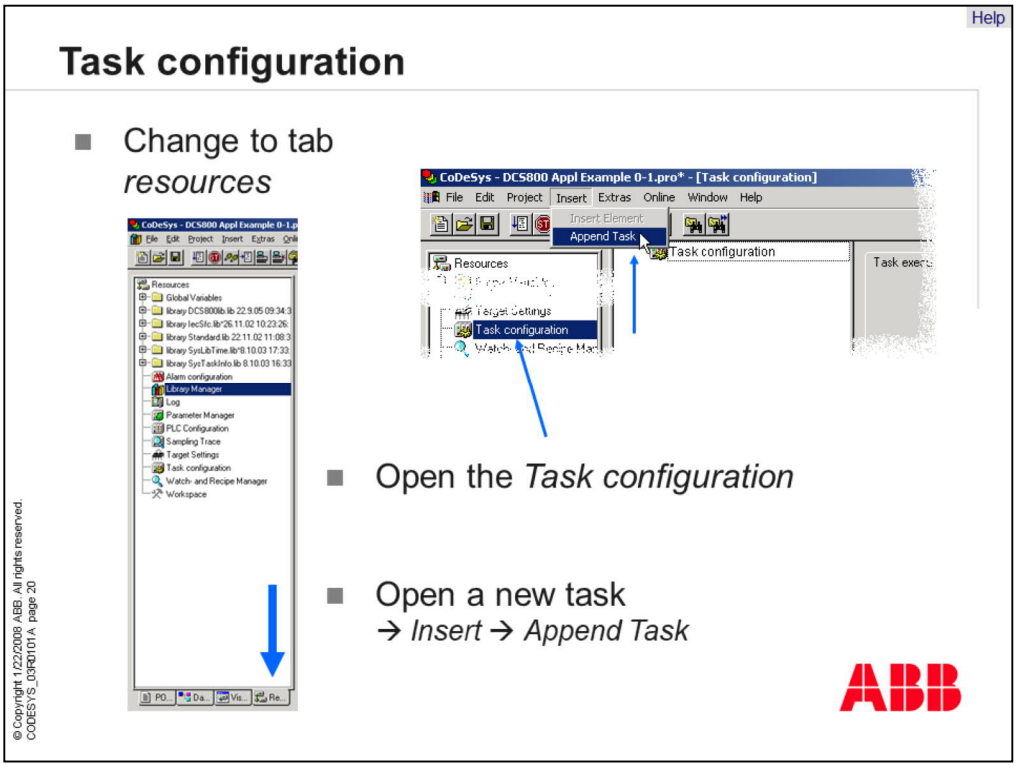


If the constant 1 is connected with *Input* “wChannel", digital output 1 is selected.

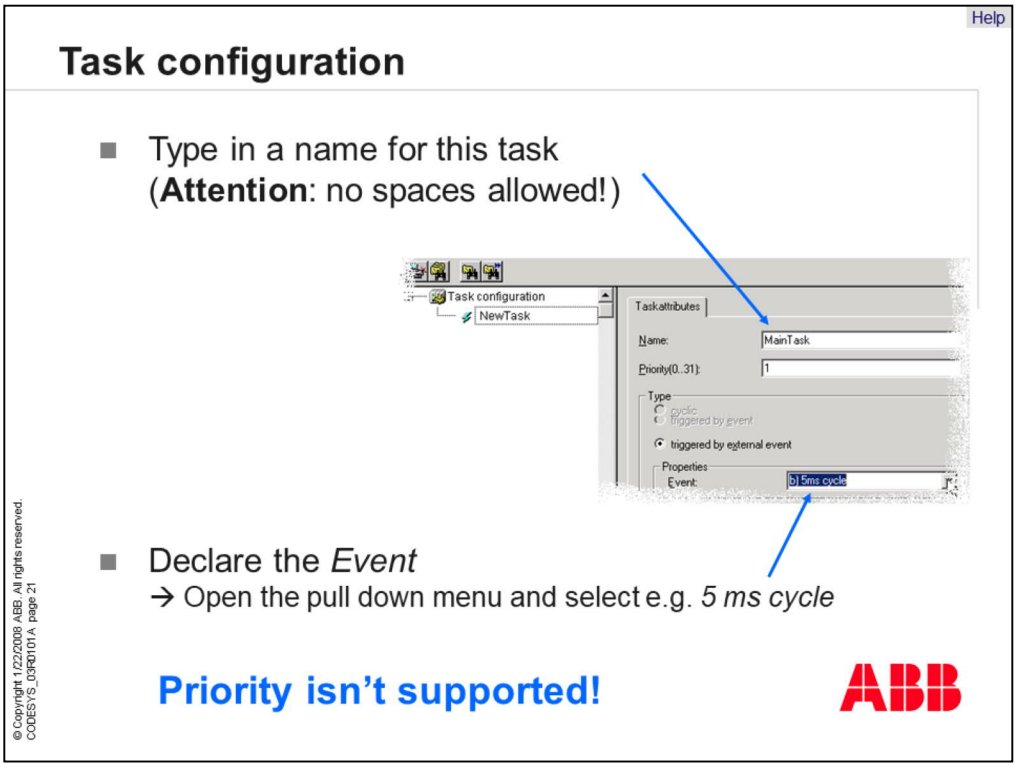
In this example:

1 for DO1

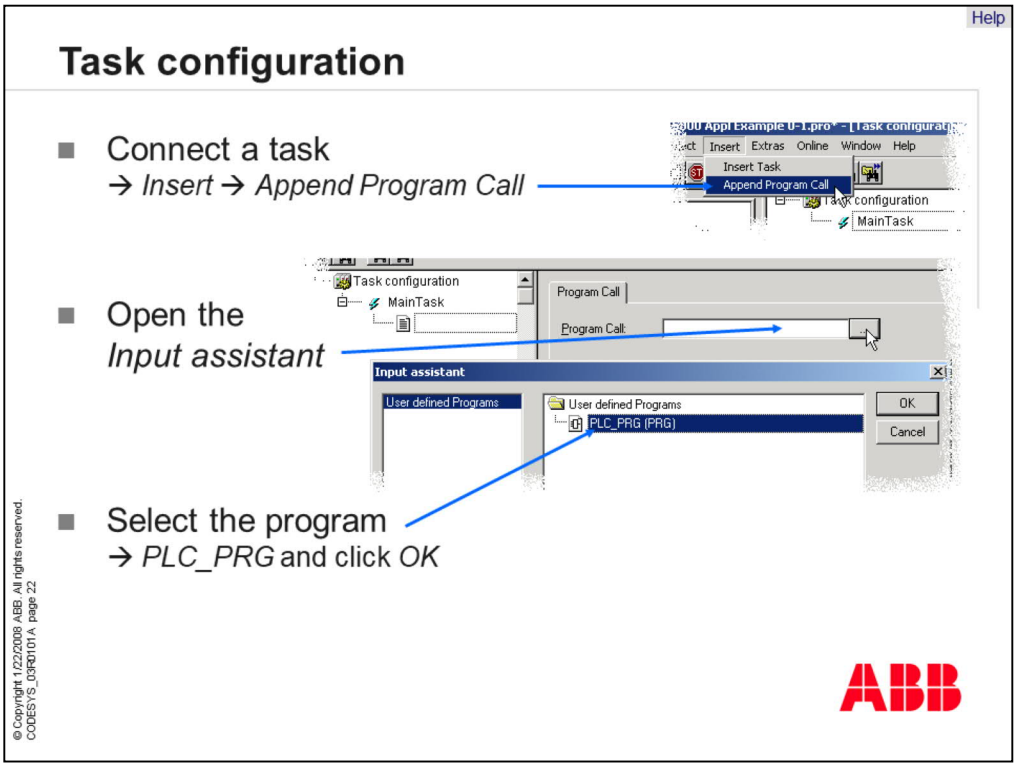
In this example we define the constant “1” for the “Channel” input. The status of “DOut” will be written to the selected “Channel”. The result can be found in parameter 7.05!



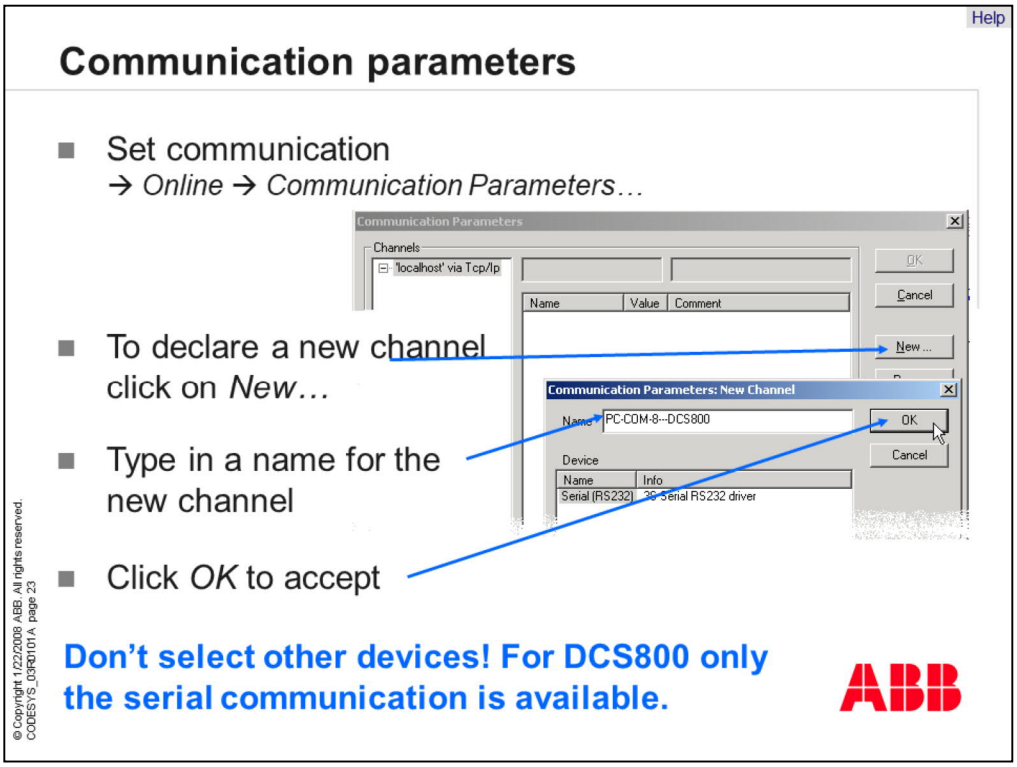
An important part of the DCS800 is the task configuration. The DSC800 only works in a task triggered mode. Switch from the main menu to the tab resources and then click on “task configuration”. Now a new window will be opened. To open a new task, select “Insert” in the menu bar and choose “Append Task”. This means that a new task will be added. It is possible to define several tasks for more than one program.



In the "task attributes" you will find several settings to configure the task cycle. The next step is to define a name for the task setting. All tasks are externally triggered events from the DCS800. Only the task cycle can be selected! It's possible to choose time cycles between 5 and 1000 milliseconds. Now one task is defined and can be used to trigger programs.



The next step is to add a program to the task configuration. This can be done by selecting “Append Program Call” in the “Insert” menu. After doing so a new window will be opened. You can add a program call to a task by using the input assistant. In this case there is only one program existing, so it has to be selected.

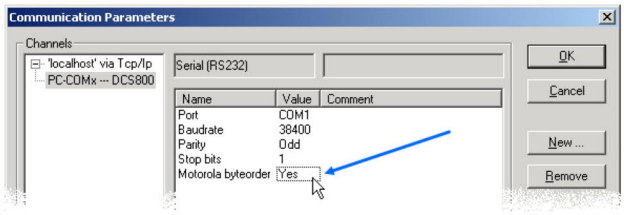


To set up the connection between the PC and the drive system, the communication parameters have to be defined. You can find this part in the menu “Online” under “communication parameters”. If there is no existing communication channel available, click on “New” and declare the new channel. Type in a name for the new communication channel and select the device for DCS800, called Serial (RS232) driver. After that, click “OK” to apply the settings. These settings are saved on the PC for the next exercises.

Help

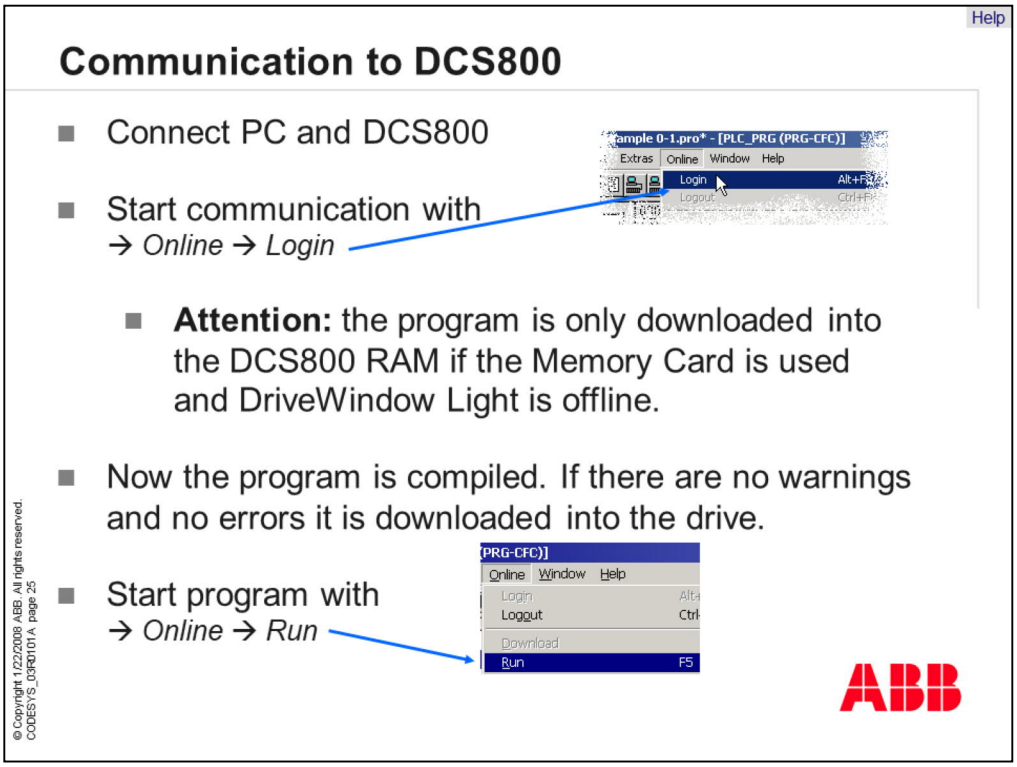
Communication parameters

- Set the communication parameters like this:

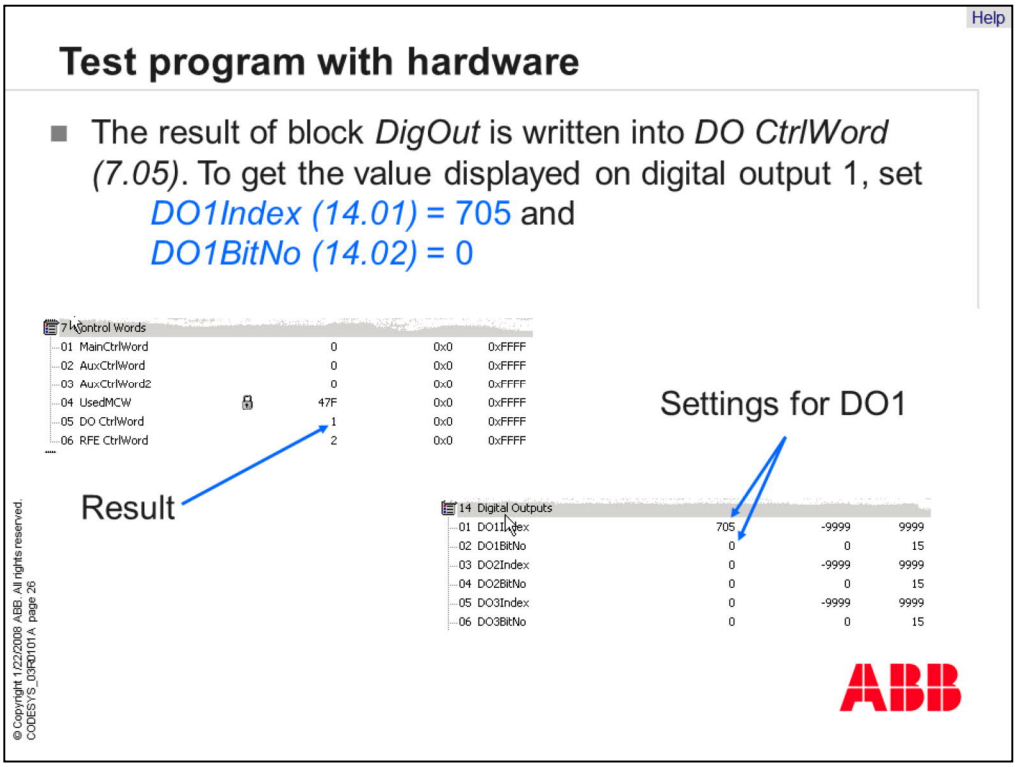


- Change the port, if *COM1* is not used
  → double click on *COMx* to change the COM port
- Change the *Baudrate*, if applicable
  → double click on *Baudrate* to change the value
- The *Stop bits = 1* and *Motorola byteorder = Yes*
- Click *OK* to apply the settings

Select the COM-port that is used for communication. If you don't have a serial port on your PC, please use a PCMCIA adapter.

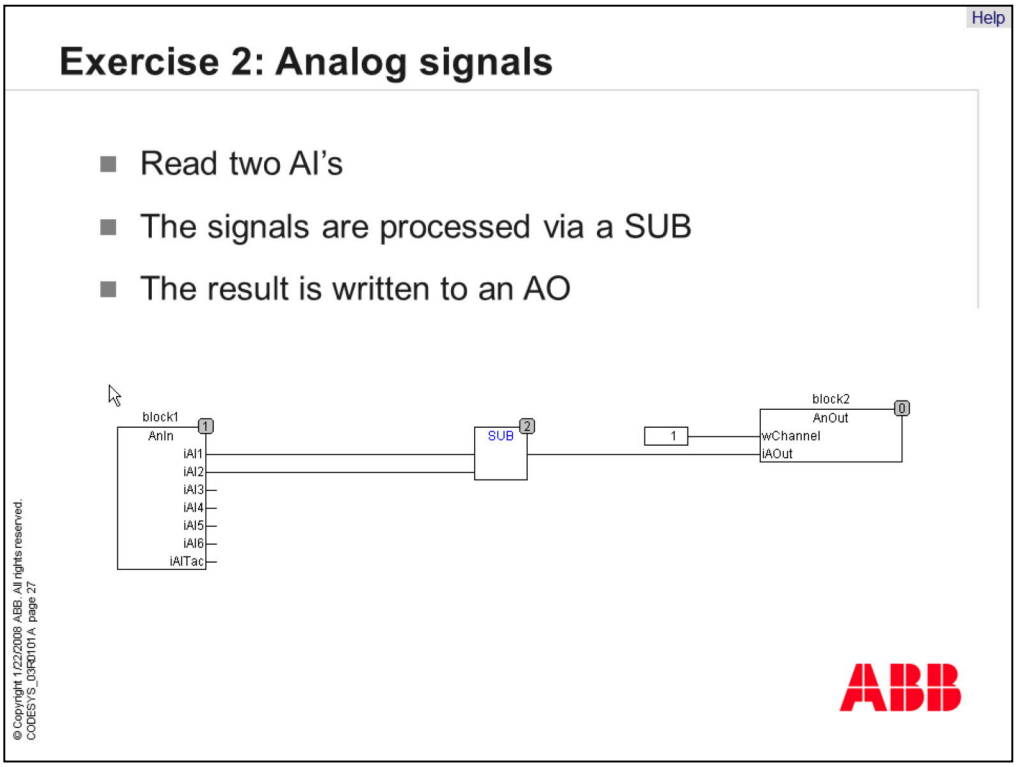


The next step is to connect the PC with the DCS800. In the menu “Online” click on the item “Log On” and the project will be compiled. After a successful compilation without any errors and warnings, the code will be downloaded to the DCS800 RAM. Afterwards activate the application program by clicking “Run”. Note, that the application program isn’t saved on the memory card, yet. This have to be done in another step.

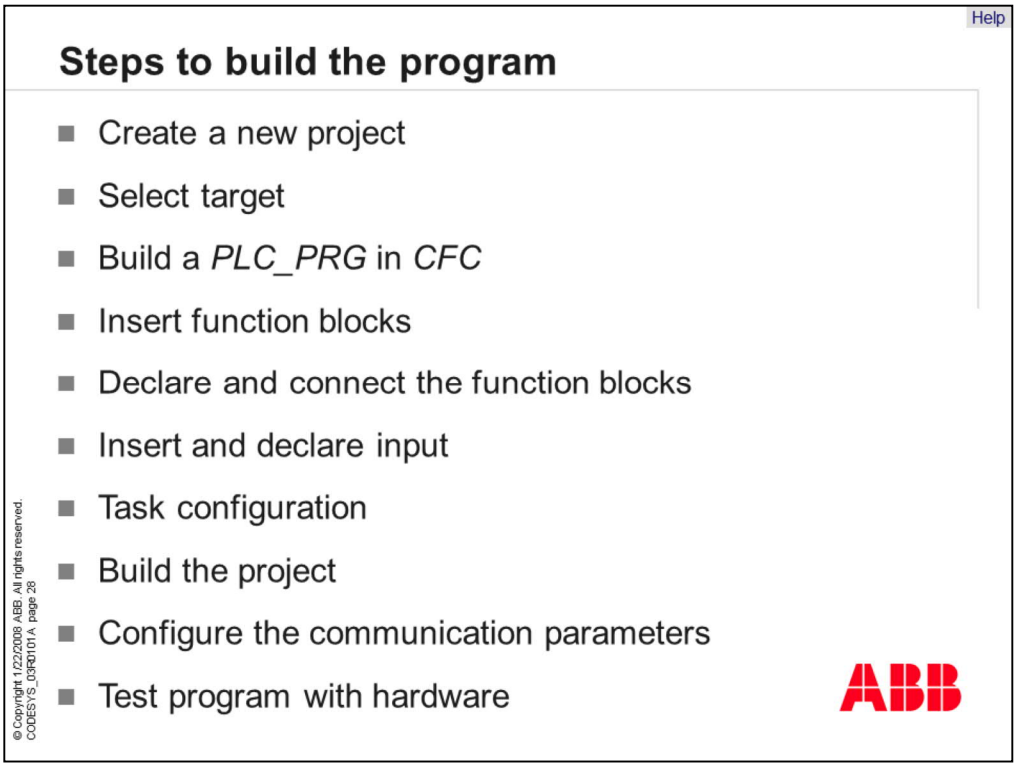


Note that the result of function block “DigOut” will be written to the digital output control word in parameter 7.05. Bit 0 accords to digital output 1 and so on. To avoid writing two sources to one sink, the control word must be connected to the physical output in group 14. In this exercise digital output, one is used and that means that bit zero includes the boolean result.

For this exercise we must write “705” in parameter 14.01. In parameter 14.02 must be written “0” to select bit 0. With this settings, the result of the application program will be sent to the correct physical output.



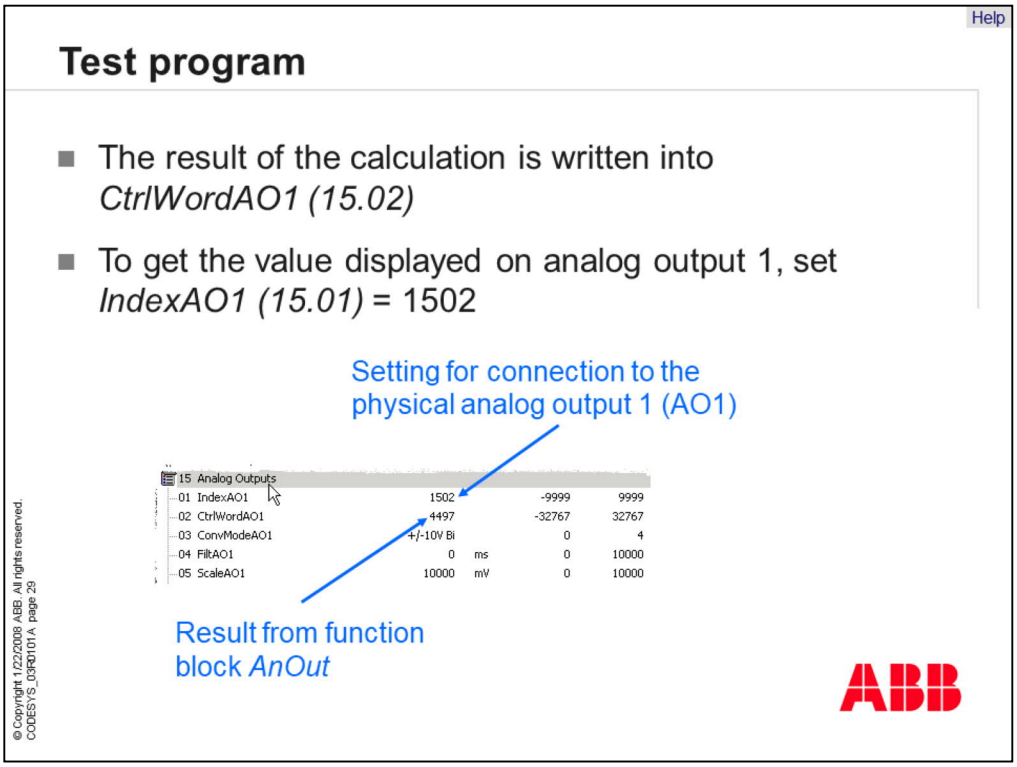
The second exercise is a combination of two analog signals of the drive's analog inputs. Function block "AnIn", is a shortcut for "read analog inputs" It transfers the voltage at the analog inputs to an integer value which accords to the scaling of the drive. In this exercise, the value of analog input one should be subtracted from the value of analog input 2. The result is written to analog output 1 of the drive. This functionality gives the function block "AnOut" which characterizes the write to analog output function.



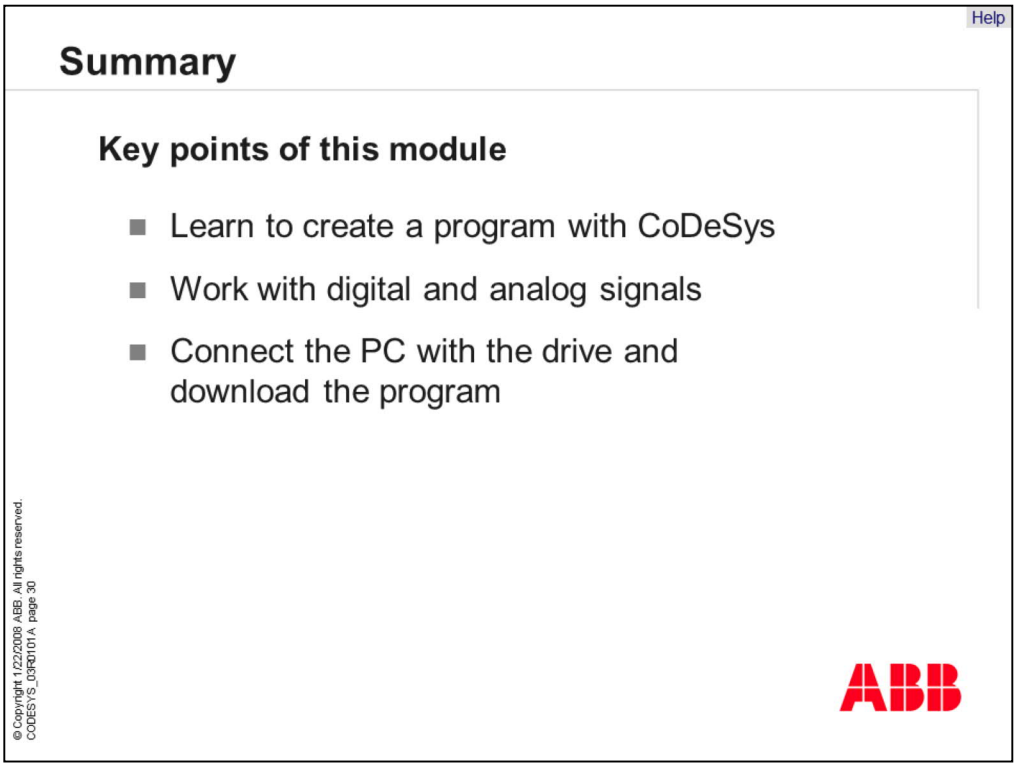
The steps to create the program are the same as in the last exercise.

The first step is to create a new project. Select the target DCS800 and build the program in programming language CFC. Insert the function blocks and declare and connect the blocks. Set the task configuration and configure the communication parameters.

Test the program with the DCS800 hardware if the program has been correctly downloaded.



A function unique to the DCS800 is the output configuration. Function block “AnOut” does not write directly to the hardware. Mainly because problems can occur if two sources are written to one sink. To avoid this, the result of the function block will be written to a “control word”. If you want to connect control word with the physical output, there must be a connection between the output index and the control word in group 15. This means that in this exercise the control word “1502” must be written to parameter 15.01.



In this module you should have learned how to create small programs with the DCS800 Control Builder and how to work with digital and analog signals. Once the program is ready, you should also know how to connect the PC via “RS232” cable with the drive and how to download the application.

Help

Additional information

- Links to related information
  - 3S-software.com
  - DC-Drive-News (Intranet)
- Additional references
  - Application Manual (3ADW 000199)
  - Firmware Manual (3ADW 000193)
  - Hardware Manual (3ADW 000194)
  - Training Material

Help

Glossary

- **CoDeSys**
  Controller Development System (software tool)
- **Memory Card**
  Flash memory
- **DriveWindow Light**
  Software Tool for commissioning and maintenance using AC/DC
- **Target**
  Interface between Drive and CoDeSys tool
- **Control Builder**
  Whole system with software and hardware
- **PLC_PRG**
  Main program which is used in all applications
- **POU**
  Program Organization Unit
- **Library**
  It includes function blocks which are given or designed by other users


Power and productivity
for a better world™

Help

Status page CODESYS_03R0101 (to be deleted in the final version!)

Commercial things	Units	Price per unit (Euro)	Price total (Euro)	responsible
Proofreading Words (number)				
Audio recording (time in minutes)				
Audio synchronization and adaptation (h)				
		Total:		

Progress				
	Preparation		approved	
Step	Date	Name	Date	Name
A	22.1.2008	J. Thoma	XXX	XXX
B	28.1.2008	R. Hoercher	29.01.2008	J. Thoma
C	30.01.2008	J. Thoma	XXX	XXX
D				
E			XXX	XXX
F				
G				

Note:

Fill out the white fields in the form after each step. Approval is from ABB!