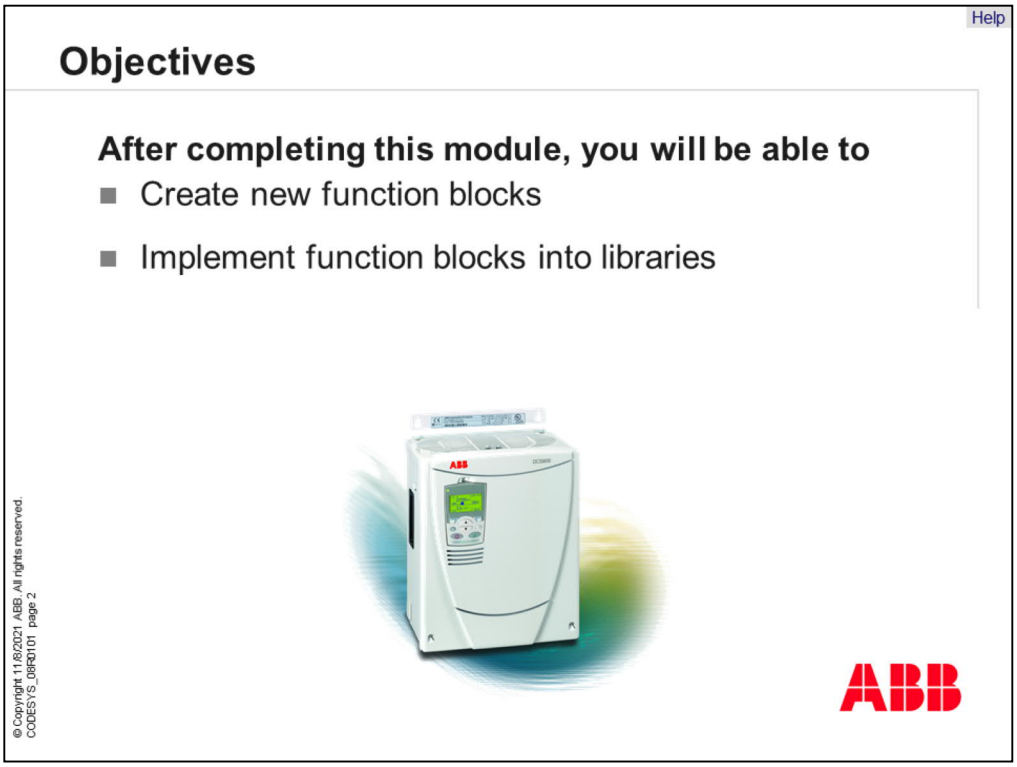
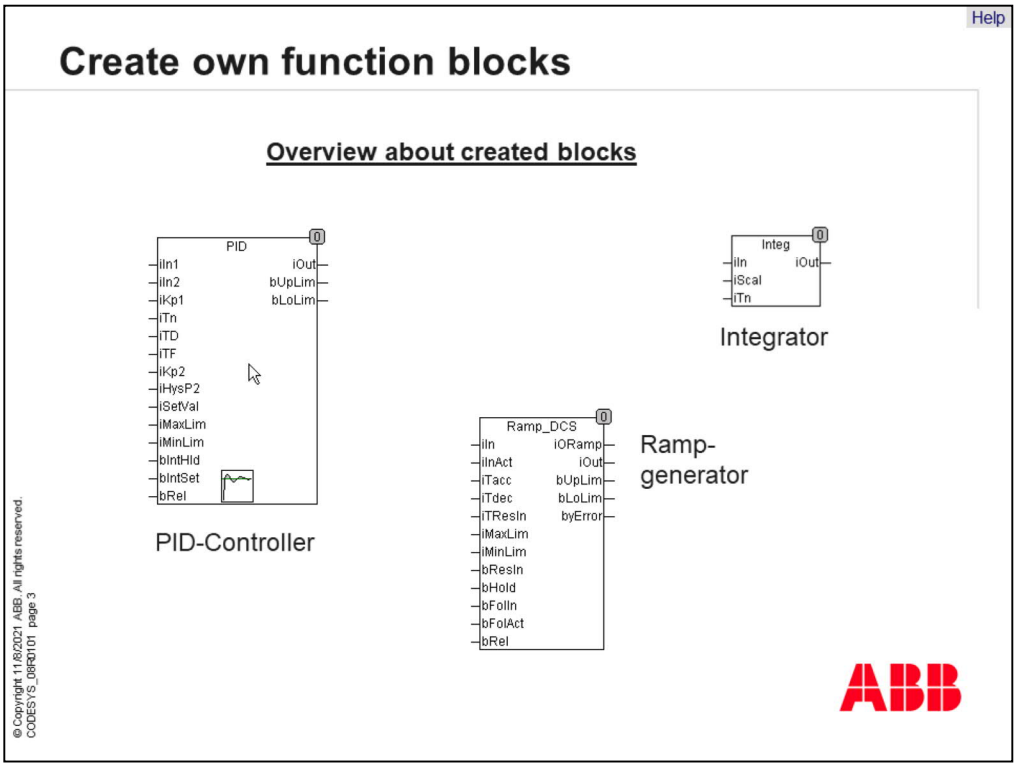
Welcome to the CoDeSys Function Blocks and Libraries training module for the DCS800, ABB DC Drives.

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.

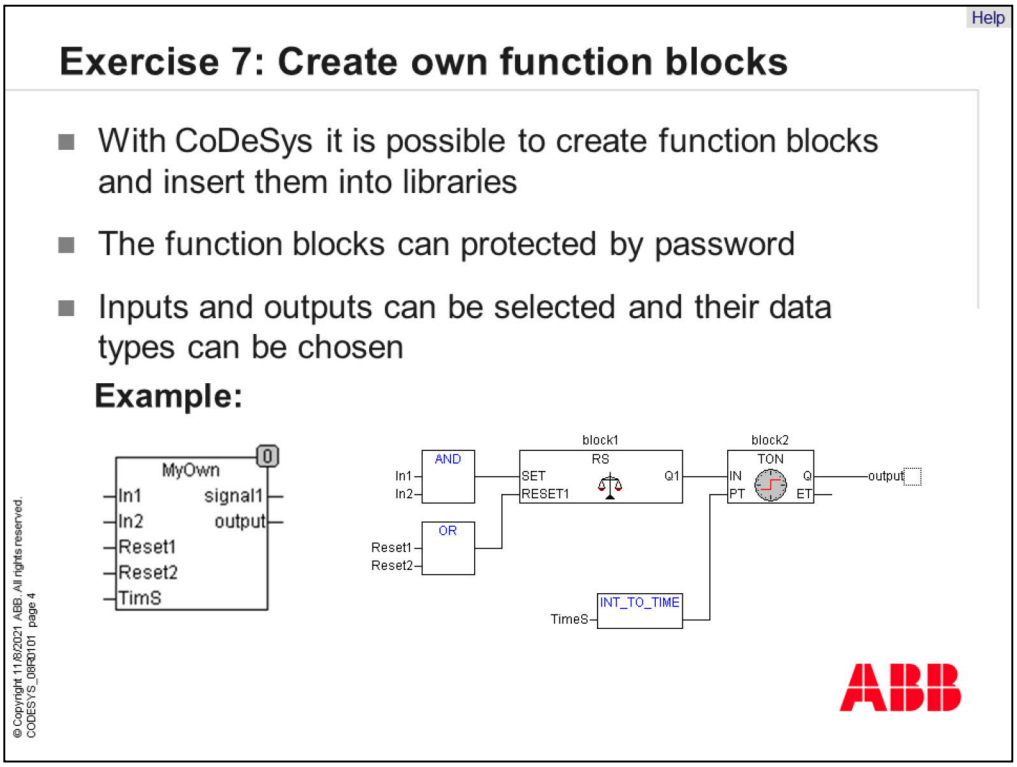


After completing of this module, you will be able to

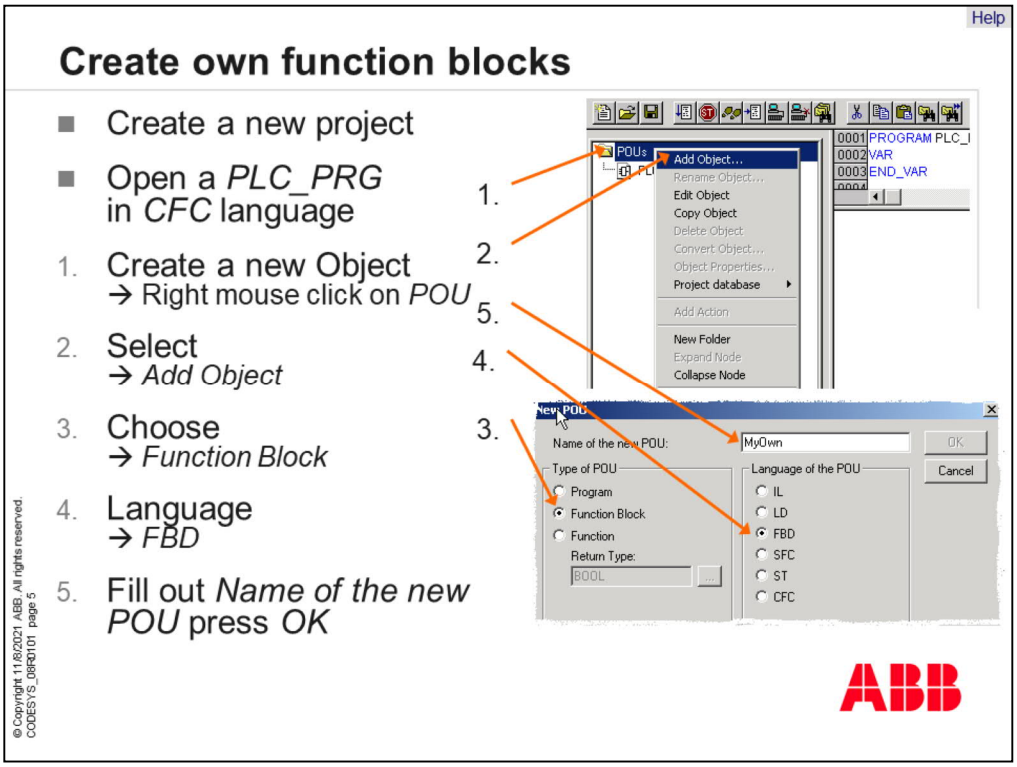
- Create new function blocks and
- Implement function blocks into libraries



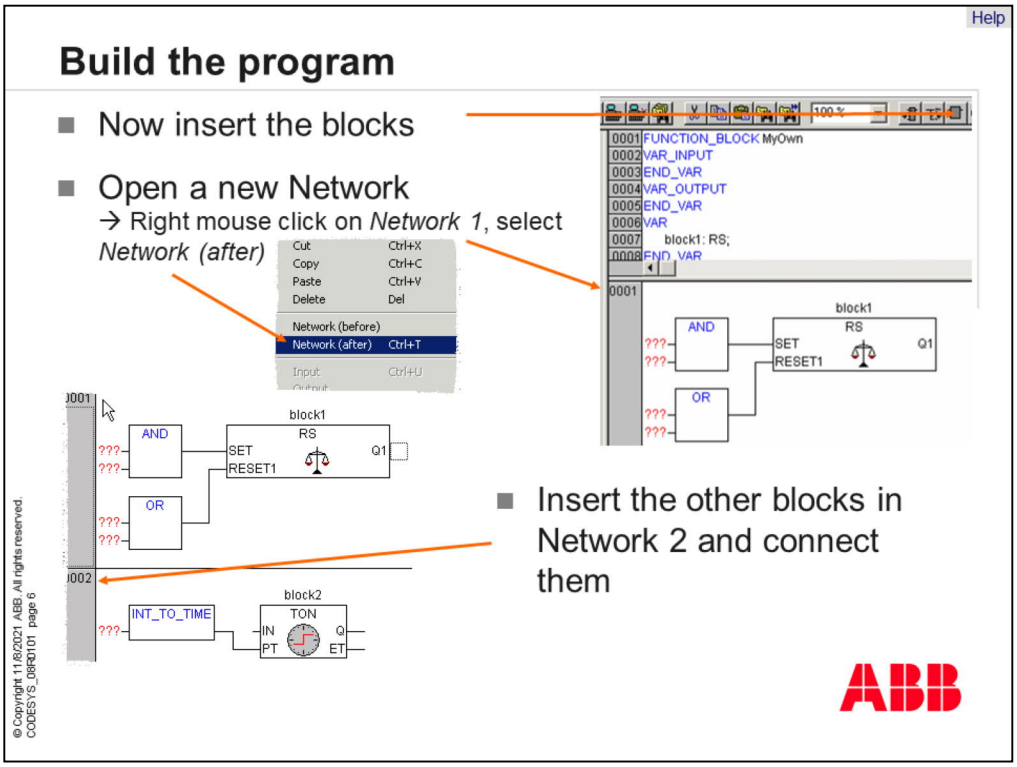
There can be an idea to put several functions into function blocks to hide know how. In CoDeSys it is possible to create own function blocks and put them into libraries. Further the user can define pictures for this blocks, like PID-controller. This function blocks are integrated in a library and can protected with a password. In this case the program code isn't visible for other users.



Let's do a little exercise. In the next exercise we build a new function block called "MyOwn". This block includes function blocks and operators from the CoDeSys standard library. The difference between programs and function blocks are that the inputs and outputs must be defined.



Next steps to build the new project are to add a new object and select the “Program Organization Unit” “Function Block”. In this exercise we use programming language “Function Block Diagram”.

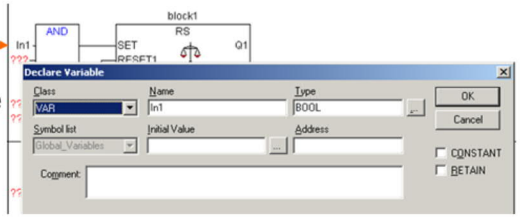


When the new network is designed, we have to put in the several function blocks. If Network 1 is ready, we add a new network after the first one and put in the other blocks. It will be avoided to split the whole program in several networks. So, it is easier to document the networks and get an overview about the whole program.

Help

Input variables

- Name the variables (e.g. *In1*)
- Click *Enter* (↵) and the *Declare Variable* Window is opened



- Declare the variable

  Class: VAR_INPUT (FB input)

  Name: In1

  Type: BOOL

Class	Name	Type
VAR_INPUT	In1	BOOL
VAR_INPUT	In2	BOOL
VAR_INPUT	Reset1	BOOL
VAR_INPUT	Reset2	BOOL
VAR_INPUT	TimeS	Int

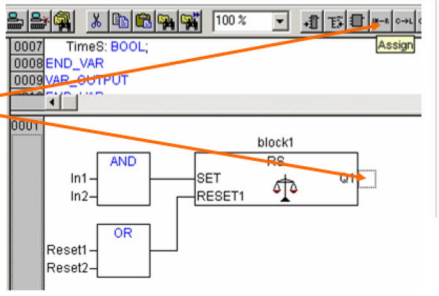
- When the settings are correct, click *OK*
- Do the same with all the other inputs

Next step is to declare all variables. This point is important because it must be defined which variables are inputs, outputs or internal variables. For our example we declare the variables like in the table.

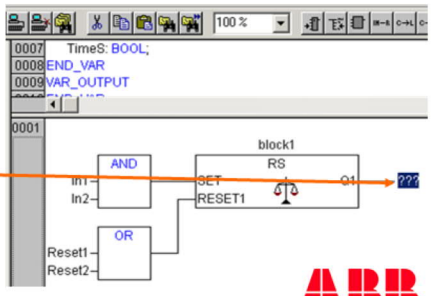
Help

Output variables

- To declare outputs, mark the output and create an assign



- At the output appear question marks



In programming language “Function Block Diagram” the first output at a function block has no connecting point. So we have to add a “Assign” and question marks are shown on the output. Now a variable for the output can be defined.

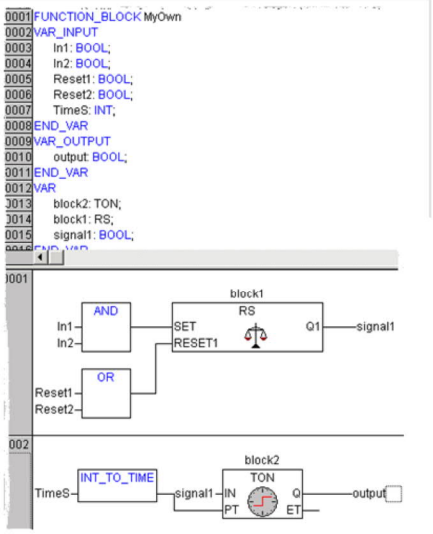
Help

Output variables

- Declare all outputs like in the table:

Class	Name	Type
VAR_OUTPUT	output	BOOL
VAR	signal1	BOOL

- Finish the program like in picture on the right!

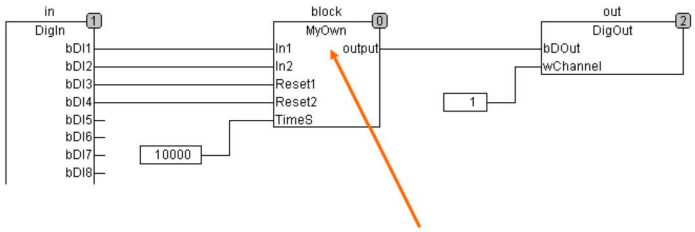


We have 1 output of the new function block. Now this variable must be declared as output variable. The picture on the right side shows the several variables for the function block. We have input variables, output variables and internal variables.

Help

Call the own function block

- Change to *PLC_PRG*
- Insert the following three blocks:



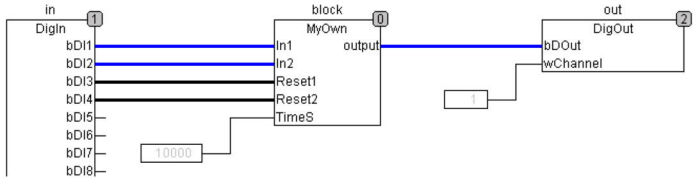
- The box in the middle is the own function block
- The other blocks are system blocks for DCS800

When the new function block called “My Own” is ready we can work with them. So, we change to the POU “PLC_PRG” and build an application like in the picture. Now the project is ready, and we can test it.

Help

Test the program

- Build your project (F11)
- Configure the communication to the DCS800
- Set the task configuration
- Login and test your program

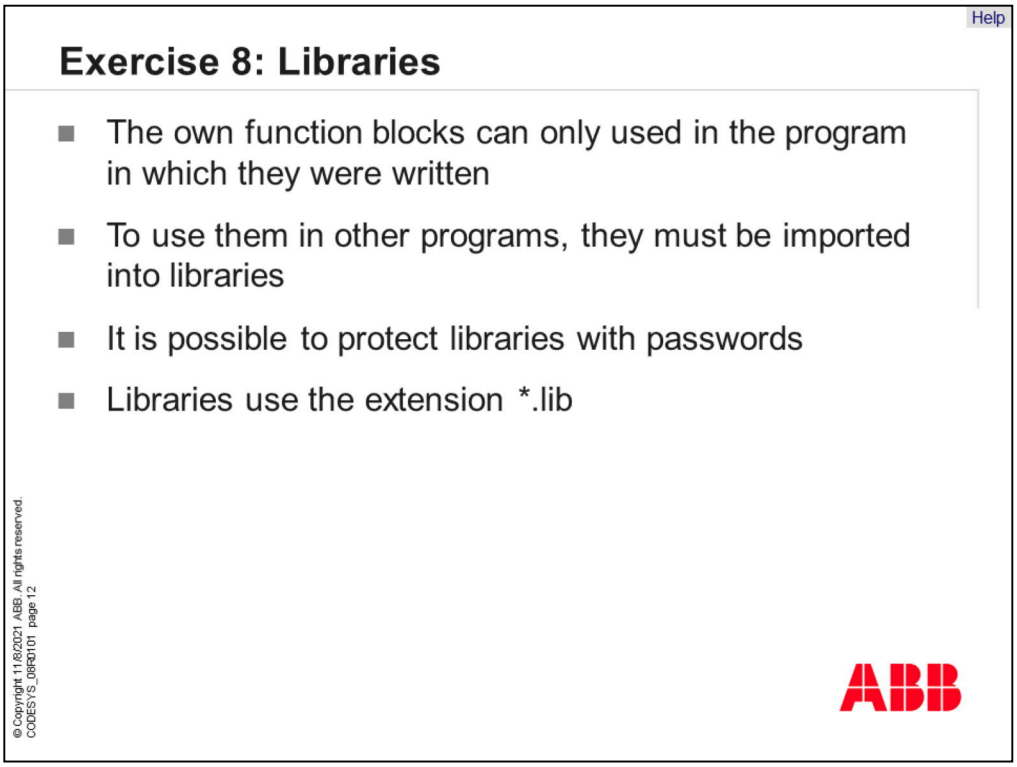


Correct function:

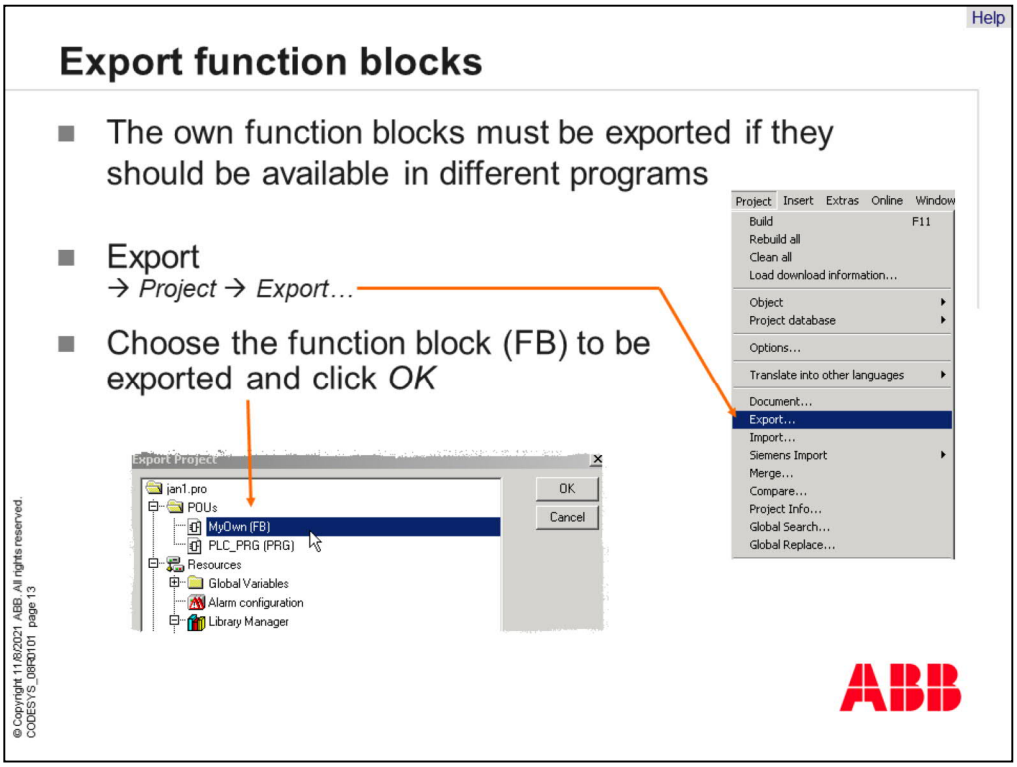
If DI1 AND DI2 are TRUE, then after 10 seconds the output is TRUE
With DI3 OR DI4 you can Reset the output (DO1 = FALSE)

Build the project and configure the communication channel to DCS800. Set the task configuration and login with the program.

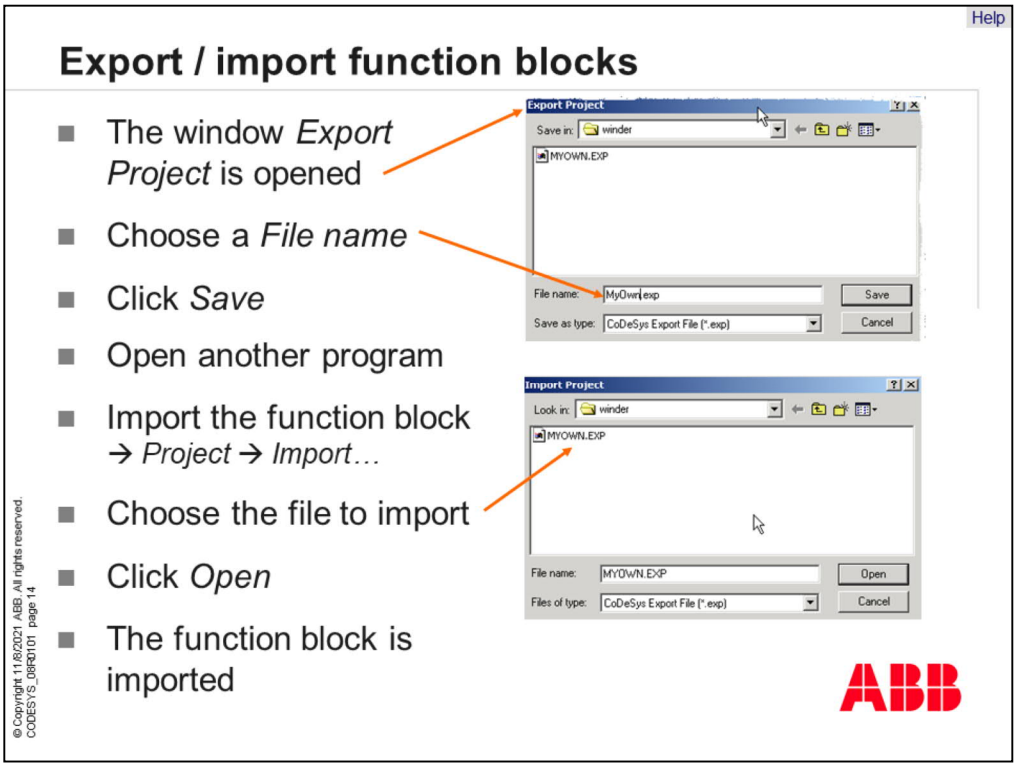
The correct function of this application is a delay of 10 seconds if digital input 1 and 2 are set.



The next exercise deal with library handling. Own function blocks can only used in the program in which they are written. To use them in other programs, they must be imported to libraries. It is possible to protect libraries with passwords.



There can be a need to use this function block in other projects. This is possible when the function block is exported to the other project.



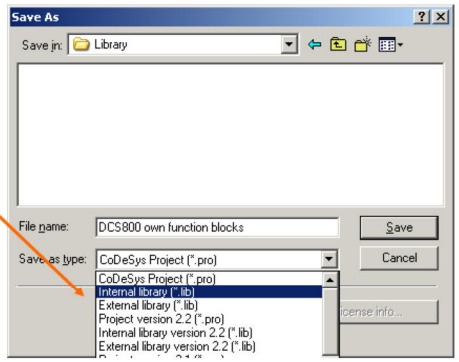
To export function blocks, define a name for the export file, please. Import of function blocks which are saved into other projects is very easy.

Open another program and click import to add the function block.

Help

Create libraries

- Create a new library
  → *File* → *Save as*
- Select library type
  → *Internal library*
- Chose a *File name* for the library
- *Save* the library

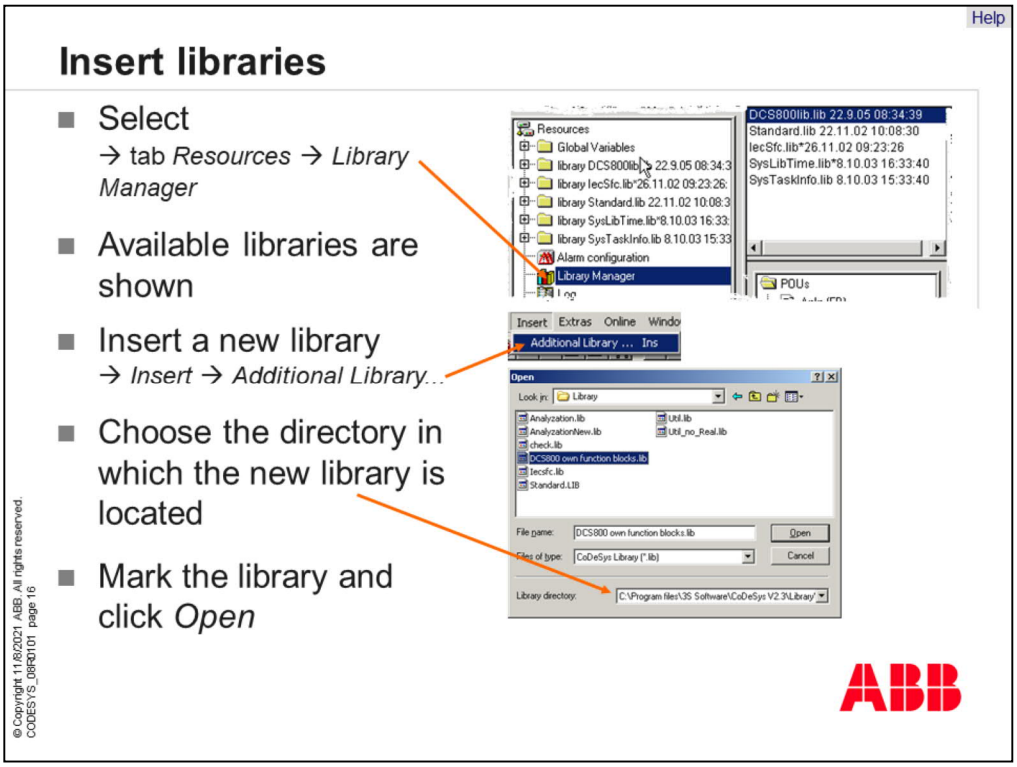


Note:

Internal library: libraries which are written with the five IEC languages

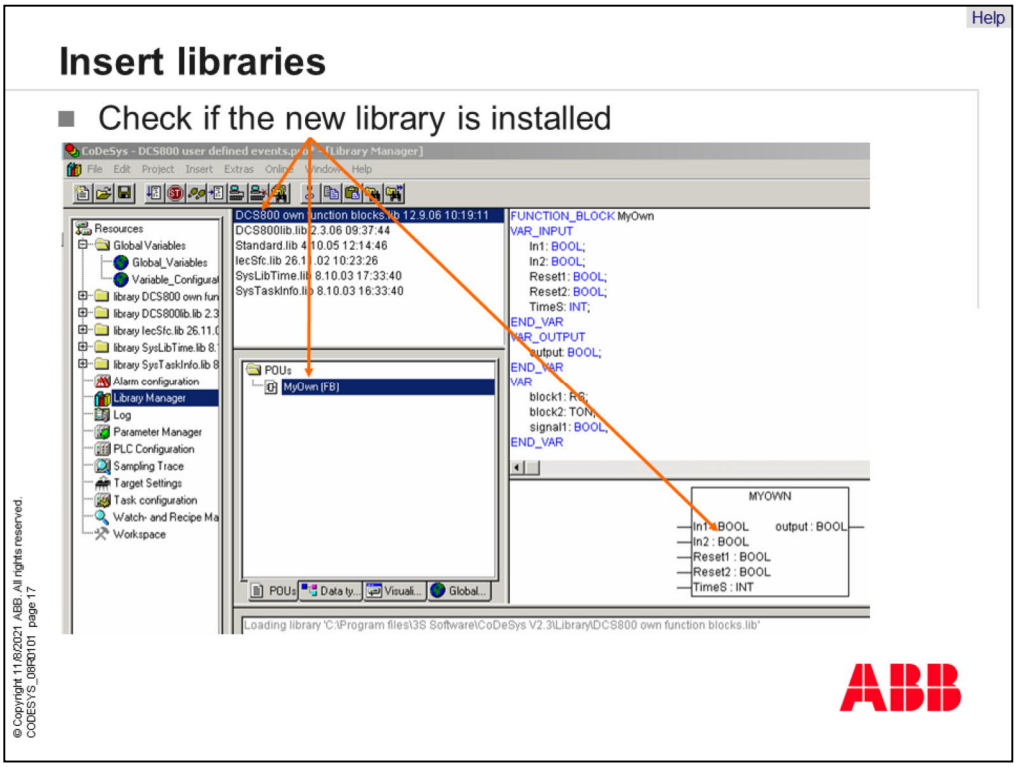
External library: libraries which used other programming languages (e.g. C)

If a new library should be created click to menu “File” the item “Save As” and select “Internal library”. Now all function blocks which are included in the project, will be saved in a new library. It isn’t possible to save programs in a library structure. Only function blocks are inside a library.

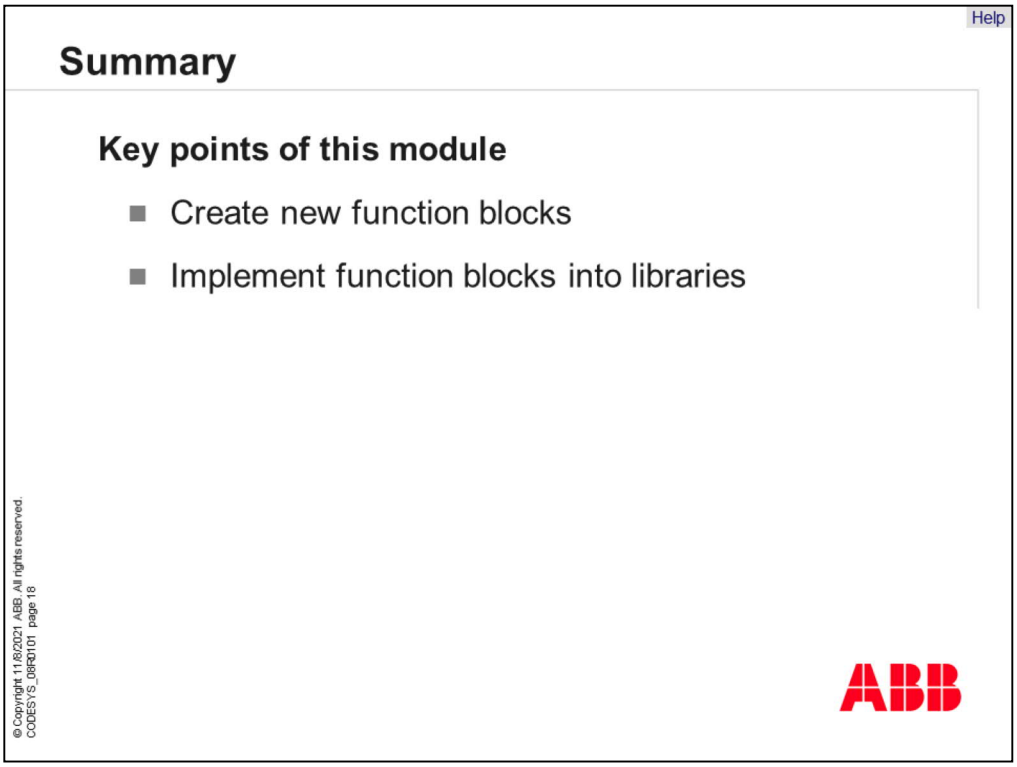


The next question is how libraries can be embedded to projects. For this case you can find the “Library Manager” in CoDeSys. Insert libraries with “Library Manager” when you click to “Additional Library”.

The library manager shows you all available libraries like the “DCS800 Library” or the “Standard Library”.



If the new library is correctly installed, you can find the library name in the directory and see the available blocks.



Key points of this module are to create new function blocks and implement them into libraries.

Help

Additional information

- Links to related information
  - 3S-software.com
  - DC-Drive-News (Intranet)
- Additional references
  - Application Manual (3ADW 000 199)
  - Firmware Manual (3ADW 000 193)
  - Hardware Manual (3ADW 000 194)
  - 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

Thank you for your attention. You may now go ahead and move on to the next unit.