Welcome to the CoDeSys basic 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 this module, you will know the basics of the “DCS800 control builder” and the features and limits of the drive system.
The DCS800 drive includes a programming opportunity called “DCS800 Control Builder”. The DCS800 “Control Builder” is based on the CoDeSys software tool in combination with the drive. CoDeSys software is based on the international standard IEC 61131 and developed by the software manufacturer 3S-software in Germany. The “Control Builder” package also includes the DCS800 target which is needed to establish communication with the drive system. The communication interface is a transducer from the software to the physical cable which must be connected with the drive. This complete “DCS800 Control Builder” package is needed to start with application programming.
CoDeSys programming is based on the international standard IEC 61131. The idea is to provide a standardized framework for the programming of control systems. In the years since the beginning of PLC programming there are several programming languages and standards. All of these well known and modern programming languages are integrated in the CoDeSys software.

The PC software “CoDeSys” is independent of the PLC manufacturer. This makes it easy to learn how to program a new PLC if you are familiar with CoDeSys from another PLC manufacturer. The common platform CoDeSys is advantageous for those working with several PLC’s because only one software package on the PC is needed for different PLC systems.
The standard IEC 61131 also specifies the hardware configuration of a PLC system. This is the electrical and mechanical construction of the PLC. Another important thing is the manufacturer documentation and the test procedure for a PLC. Basically PLCs are designed to control machines or processes in an industrial environment.
CoDeSys software includes 6 programming languages which are precisely defined in the IEC standard. The first language is “Instruction List” which is comparable with assembler programming. In America “Ladder Diagram” is often used. It’s like an electric circuit diagram. The most used programming language is “Function Block Diagram”. With this language the user has multiple blocks which can be connected. “Sequential Function Chart” is a phase diagram. It starts with the first step and runs to the next one if the conditions are met. “Structured Text” is a high level programming language like “Pascal”. It’s purely text programming without any pictures or graphical design. “Continuous Function Chart” is a fully graphical tool and is similar to function block programming. The Difference to function block programming is the free adjusting of the blocks on a sheet.
An example of the programming language “Function Block Diagram” is shown the graphic above. This is a programming language based on boxes placed in networks. It's possible to put several function blocks into networks and to connect them to each other. The execution order is from network 1 to the last network. The connections between several networks are done via variables. Constant values can also be connected directly to the inputs.
The programming language “Continuous Function Chart” is similar but not exactly the same. The user has an empty sheet to put in several function blocks. Then all connections can be made via wires between the function blocks. The execution order is normally from the left to right and is shown by numbers on the function blocks. It’s necessary to set input and output blocks to connect variables and constant values.
Structured text is a “high level” text programming language like Pascal. It's for advanced users who can handle high level programming languages. Functions like loops and executions are implemented, if conditions are met. Based on processing time “Structured Text” and “Function Block Diagram” are the fastest programming languages. For CoDeSys beginners it is recommended to start with graphical languages because they are easier to understand!
The IEC 61131 also defines the structure of a project. That means the systematic splitting of the whole program into several small parts to get a structured program. Inside CoDeSys there is a difference between the types of blocks.

Operators define logical blocks like an “AND” or “MULTIPLIER” functionality.

Functions have only one output and no memory functionality.

Function blocks have full functionality with several inputs and outputs but must be declared as a program instance. That means each function block gets a name which is directly sorted to the function block!

In CoDeSys all variables must be declared with the current type and the current class. This means it’s a global or a local variable.
The most important part in CoDeSys is the desktop window. This is the main window where all functions can be controlled.

The desktop includes several areas which will be described in the following pages.

- In the menu bar you can find all available functions of CoDeSys.
- Below is the tool bar with shortcut icons for the most frequently used functions.
- The object organizer shows all programs and functions in a directory.
- The editor window helps you to build your application with function blocks or with a text-based language.
- In the message window you can find errors and messages.
Let us take a closer look at the windows.

- The “Object Organizer” lists all used programs, function blocks and functions for the current program organization unit. This is the main directory for a CoDeSys program and from here you can access each program.
- The “Editor Window” includes the actual application program with the logic or calculations. This window includes the main task of an application program.
- Above this editor window is the “Declaration Window”. In CoDeSys programs all variables and function blocks have to be declared. These names can be found there.
All menus and functions of CoDeSys can be opened in the “Menu Bar”. The handling is the same as with familiar windows programs.

An important part of CoDeSys is the “Message Window”. Here you can see all errors and the actual state of your program. A double click on the errors shows you directly the line or network with the problem.

With the “Window Changer” it’s possible to switch between program units, new user structures, visualizations and resources. The resources window is very important for modifying the settings for task configuration or libraries.
All used variables and instances must be declared in CoDeSys. Function blocks must be declared in instances. An instance is a name of a function block which defines the box and allows you to save values inside a function block. The instance of each function block can be found at the top of the block and in the declaration window.

Also, variables must be declared with the correct data type and class. Make sure that all variables have the correct data type and correspond to the program. If variables are not declared correctly, you will receive error messages.
Making connections between function blocks in several networks is possible by using variables. The variables must be from the same type and declared as either local or global. This procedure is necessary for the following programming languages: “Function Block Diagram”, “Structured Text” and “Instruction List”. 
In CoDeSys there are several types of variables. The easiest version is the local variable. This variable can only be used in one program.

Global variables are available in all programs and should be used to transfer data from one program to another.

For function blocks the input and output type variables must be used. Input variables define the inputs of a function block with the data type. The same rule applies for the outputs.
In CoDeSys there are several data types available. Standard type in DCS800 is the integer or word type because the processor is a 16-bit processor. Other types are boolean, byte, double integer or word and real types. It is not recommended to use real types because the processor load is much higher than with other types. By using real types, the program capacity decreases rapidly.
In CoDeSys several data types can be used. The data types must be selected if a variable should be created. The table provides an overview about the available data types and the working range.

Normally it is recommended to use boolean, integer and word types. Also, double integer and double word types can be calculated by the processor quickly.
An important difference between external PLC’s and the DCS800 drive system is the hardware configuration.

Normally PLC’s will be configured by using the CoDeSys software tool. CoDeSys includes a “Hardware Manager” which is used to add additional hardware such as input and output devices.

With the DCS800 drive the hardware is fixed and all hardware settings must be set in the drive’s firmware. But all inputs and outputs, as well as the firmware parameters, can be used by the application program.
The graphic shows the configuration of the CoDeSys software tool and the DCS800 converter.

In order to use the application’s functionality on DCS800, the “ABB Memory Card” must be plugged in. It includes the saving functionality and serves as a hardware dongle. Then the input and output pins from the drive can be used for application programming.

The graphic shows the complete connection configuration between the PC and the drive. On the right side of the graphic is the CoDeSys software tool which is installed on the PC. It’s connected with the drive via a serial “RS232 connection”. All sensors and actors are directly connected with the drive and communicate with the application program.
The installed CoDeSys software tool, together with the DCS800 target and library, is needed on the PC for using the DCS800 Control Builder. The software is available on the DCS800 Customer CD which is delivered with each DCS800 drive.

For an online connection between the drive and the PC, the PC must be connected via serial “RS232 cable” with the DCS800 converter. The specially formatted “ABB Memory Card” is also needed to download the application to the drive and to go online.

CoDeSys uses the same serial communication as “DriveWindow light” and the “Hitachi tool” for the firmware download. Therefore, only one program can communicate with the drive at a time. If the PC does not have a “RS232 connector”, please use an adapter.
There are several possibilities to equip your PC with a serial port if there is no physical port available. The most reliable adapter is a PCMCIA card with a “RS232 connector”. Another possibility is a “USB to serial adapter”. Please note, that a high-quality adapter is needed, otherwise the communication will not work properly.
Here are some features of the “Control Builder”:

- Creating programs in 6 available programming languages and splitting the programs into small parts to work in different task cycles.
- Connecting analog and digital signals from the DCS800.
- Connecting all firmware signals and parameters directly with the application program. Special function blocks are available to have a read and write functionality.
- Creating up to 250 new user parameters and defining the range and unit, such as the properties for these parameters.
- Creating new user events like alarms, faults and notices, to adapt the drive to the application.
- Working with several task cycles and single shot actions. It’s possible to define several task cycles for each program part.
- Function blocks are available in libraries and must be installed. There you can find arithmetic and bit operating functions as well as special winder function blocks.
- To check signals, a recording tool called “Sampling Trace” has been implemented in CoDeSys. With this tool all variables inside CoDeSys can be recorded.
The DCS800 Control Builder has a limited power range. It uses the drive’s 16-bit processor. This means that the DCS800 is only a small PLC and cannot replace a large external PLC.

We do not recommend using real types because the processor load is round about 10 times higher than with integer or word types.

The task cycle time limits the number of operations in the selected task.

The total processor load is limited by the calculation of the firmware control structure and the calculation of both application systems.
The configuration of the application system can be split into two parts. The first part is the CoDeSys development system. The other part is the runtime system DCS800.

The procedure to implement applications starts with creating part of the project using CoDeSys. After that the CoDeSys program has to be compiled. The result of the compilation is a code which can be transmitted to the DCS800 converter. If the application program is enabled, it's working. With each power-up of the drive, the application starts automatically until the application is disabled.
The development system is the CoDeSys software. It is installed on the PC and includes the following parts:

- An editor to create the application program, the compiler and the debugger.
- The software supports all 6 IEC programming languages.
- The code generator is also included to translate the program code into a code which is compatible with the processor.

This job is done directly by 3S-software, the manufacturer of CoDeSys software.
When working with CoDeSys there are many files which are important to know. The program code is in the file with the extension "*.pro". It’s only on the PC. The program code and the checksum file must be transferred to the drive. The Source-code includes data about the program. It can be saved on the "Memory Card" and then you can upload the code and look into the application program. Another important file is the target file. This file includes information about the complete application system with the processor.
The runtime system is the DCS800 CPU. All components which are part of the drive, and the firmware are part of the runtime system.

This includes the update of the inputs and outputs of the drive, the communication with the development system as well as the boot procedure after a power-on. The runtime system is also responsible for saving the program code, in this system the “ABB Memory Card” and the stand-alone system.

The runtime system is developed by ABB development.
To implement application programs to a drive system, several things must be configured in the parameter list. The digital and analog inputs and outputs will also be configured by parameters. This means that the CoDeSys programmer must also set the necessary parameters. This means an application consists of a CoDeSys program and the configuration of parameters outside CoDeSys program. It is important that the programmer has knowledge about CoDeSys programming and drive configuration. Otherwise, problems can occur between application programs and firmware.
Professional programming requires knowledge of the important tools of DCS800. “DriveWindow” software is used to set drive parameters and monitor drive signals or parameters. It is also used for fault tracing.

CoDeSys can be used to create the program, compile it and transmit the code to the drive system. Forcing variables online is also possible.

The advantage for the user is that both programs can be used simultaneously because “CoDeSys” is connected via a serial cable and “DriveWindow” via a fiber cable.
Whenever you need help with the CoDeSys software tool, you can find information in the help wizard. It describes the “tool functionality”, the operators and the “standard library”, such as errors and hints. Information about the DCS800 libraries and the DCS800 drive are not included in the help file. In such cases, please refer to the ABB documentation on Hardware, Firmware and Application.
An important document for working with applications is the “Application Manual”. The manual describes the installation of the software and the “DCS800 target” on the PC, the required hardware and the handling of the ABB “Memory Card”. The description of the libraries and the specially designed error messages of the DCS800 are also included.

Information about the standard system software of the drive is provided in the “Firmware Manual”, in which all signals and parameters of the system are described as well as the start-up of a drive.
In this module you should have learned the CoDeSys basics.
This consists of the configuration of the development and runtime system, the several programming languages and the data types. Also, part of this module is the principal handling and structure of the CoDeSys tool.
Additional information

- Links to related information
  - 3S-software.com
  - DC-Drive-News (Intranet, only ABB internal)

- Additional references
  - Application Manual (3ADW000199)
  - Firmware Manual (3ADW000193)
  - Hardware Manual (3ADW000194)
  - Training Material
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**
  DCS800 application system

- **PLC_PRG**
  Main program which is used in all applications

- **POU**
  Program Organization Unit

- **Library**
  Includes function blocks which are given or designed by other users
Power and productivity for a better world™