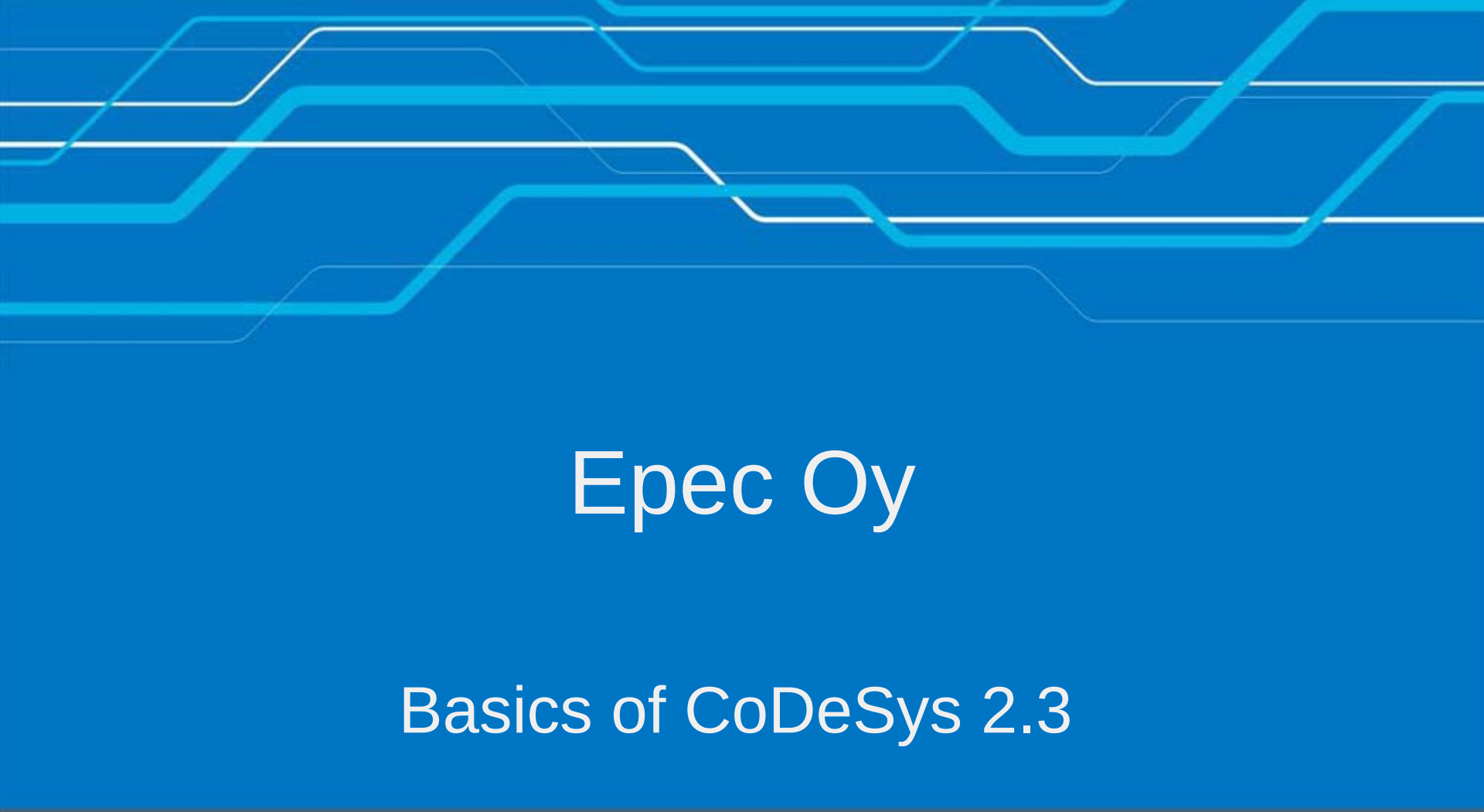




Epec Oy

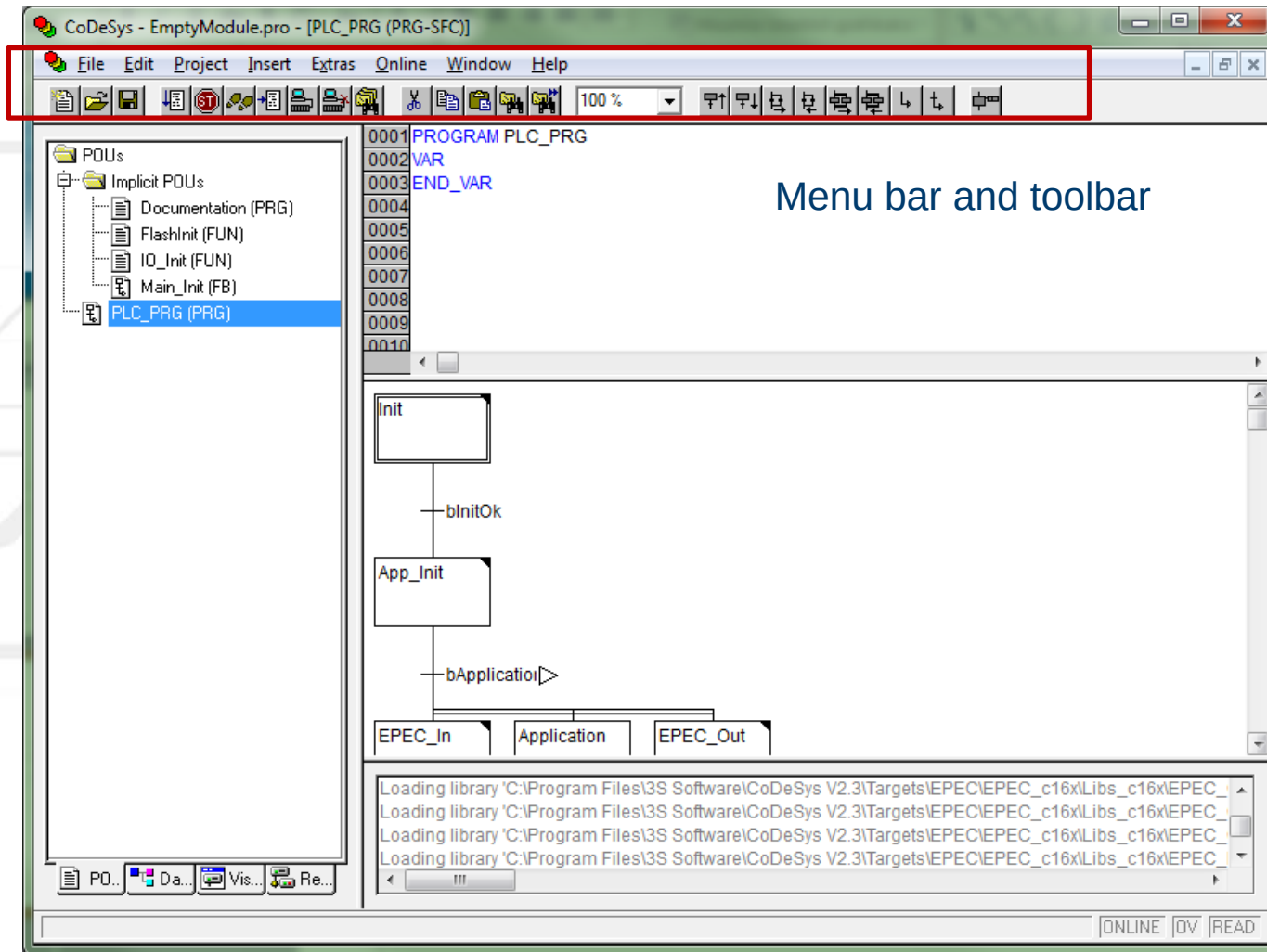
Basics of CoDeSys 2.3

The logo for EPEC, consisting of the letters 'EPEC' in a stylized, blue, blocky font.

What is CoDeSys?

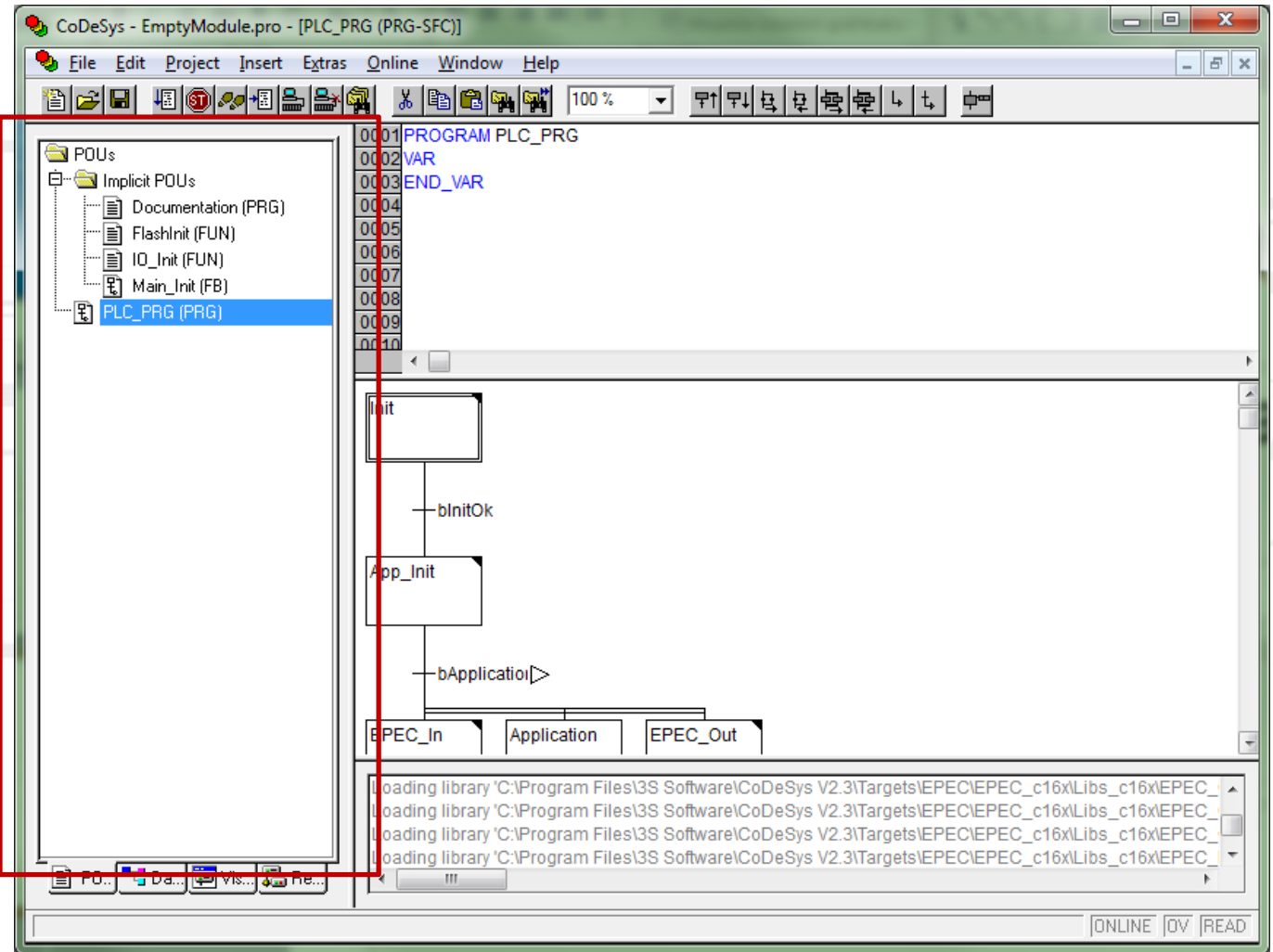
- IEC 61131-3 programming tool
- Graphical UI
- Debugging tool
- 6 different programming languages
- Only way for programming Epec control units

User Interface

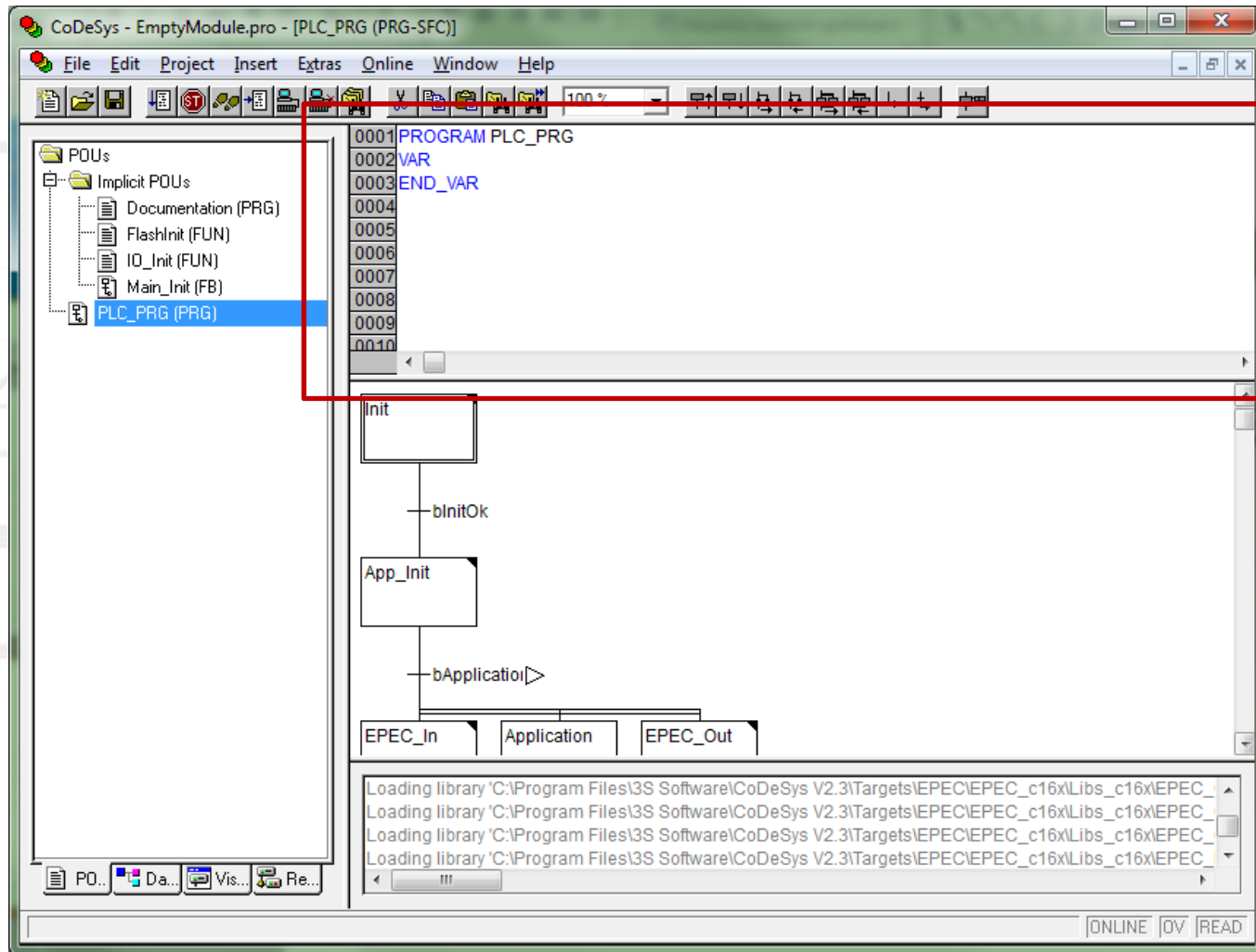


User Interface

POU's

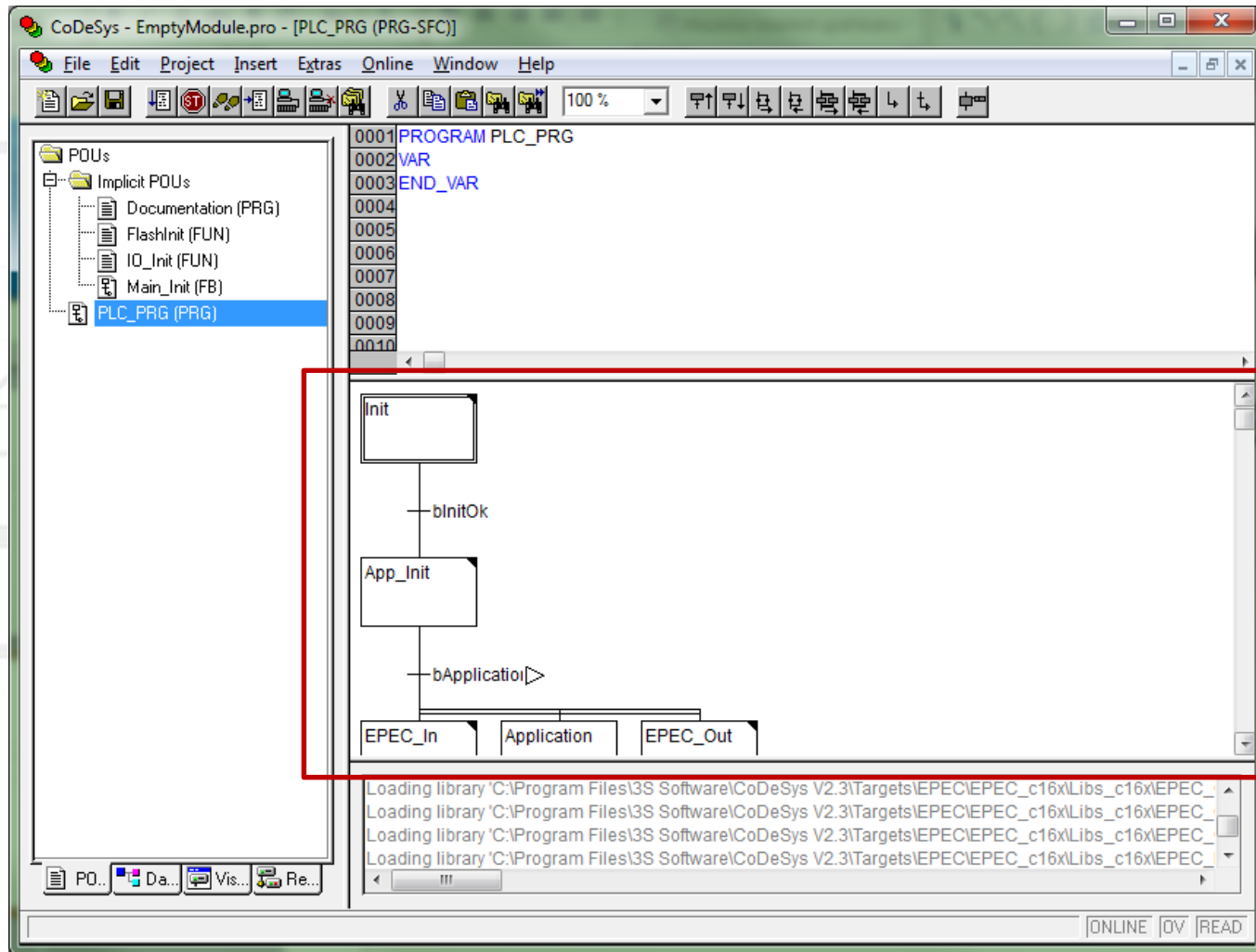


User Interface



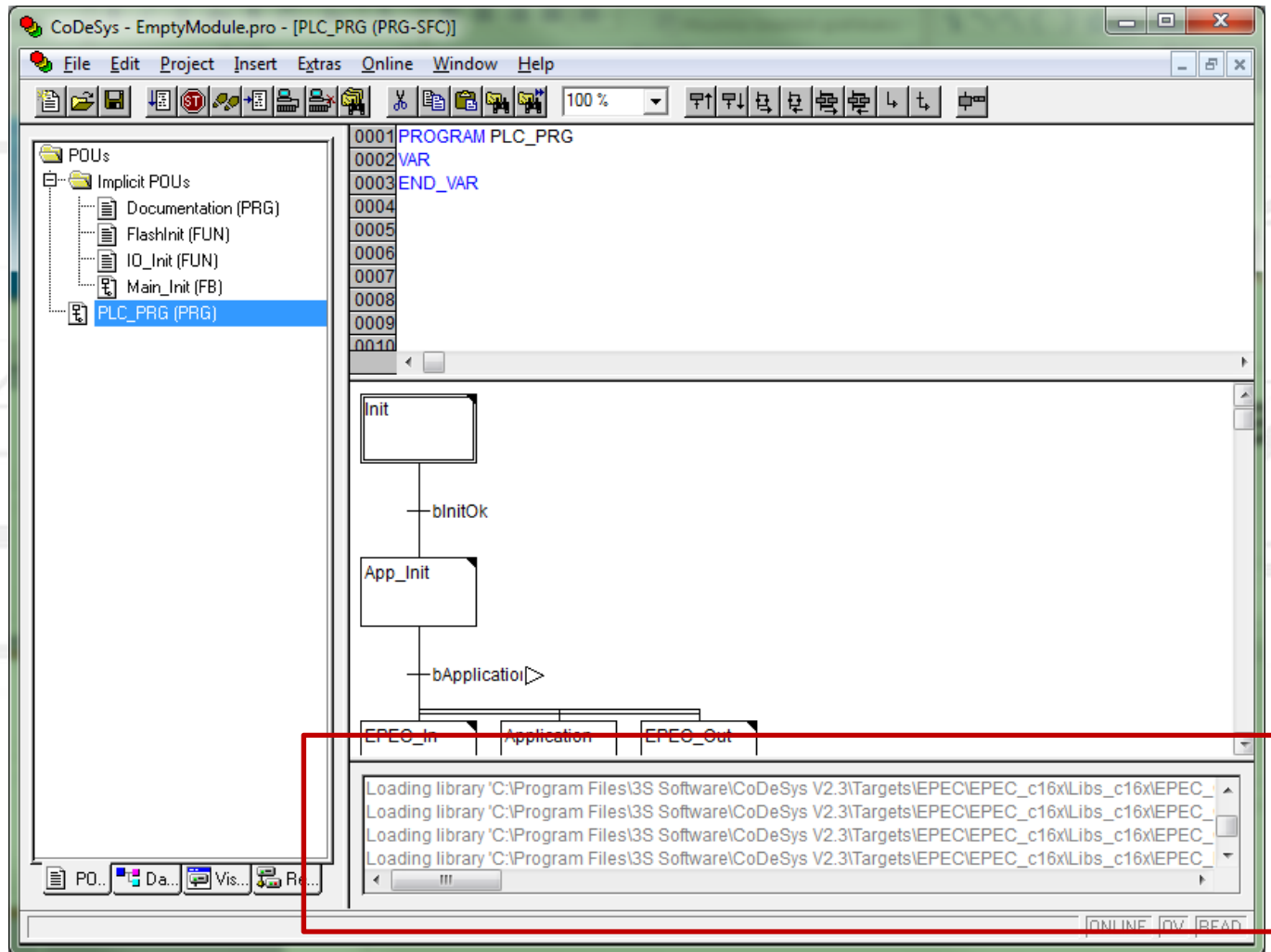
Declarations

User Interface



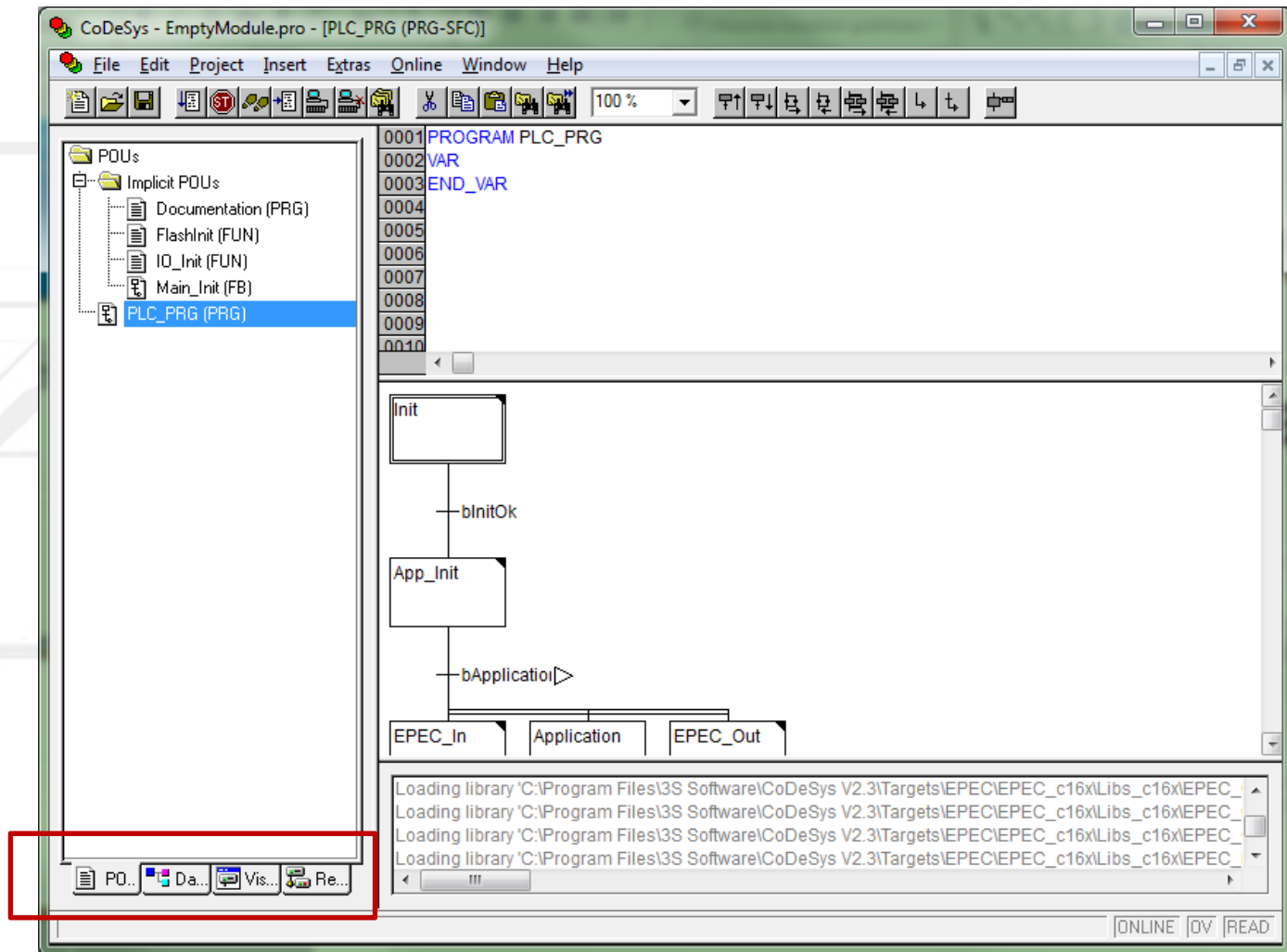
Code

User Interface



Messages

User Interface



- Tabs
- POU's
- Data Types
- Visualization
- Resources

Structure of a project

- Project
 - POU's
 - Declaration
 - Code
 - Datatypes
 - Visualizations
 - Resources
 - Global variables
 - Libraries

POU

- Program organisation unit
- Subroutine of an application
- Three types
 - Function
 - Function Block
 - Program
- Application execution starts from POU
PLC_PRG

Function

- One or more inputs
- One output
 - Name of the output is the name of the function
- Internal variables are reset on each call
- Used for calculating algorithms

Function Block

- One or more inputs
- One or more outputs
- Several instances can be created
- Each instance has its own internal variables
- Variables remember their previous state

Program

- One or more inputs
- One or more outputs
- One global instance
- Variables remember their previous state

Different programming languages

- IL – Instruction List
- LD – Ladder
- ST – Structured Text
- FBD – Function Block Diagram
- CFC – Continuous Function Chart
- SFC – Sequential Flow Chart
- All programming languages can be mixed in an application

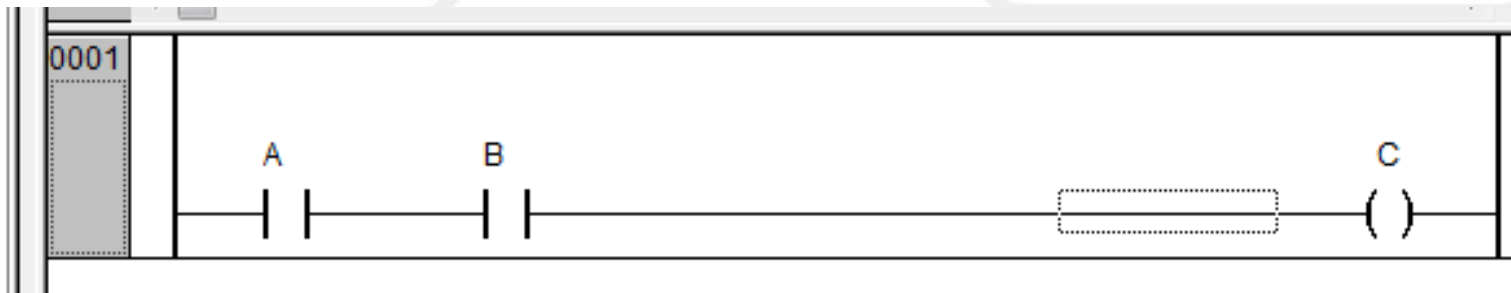
IL – Instruction List

- Textual language
- All operators work on a special register (accumulator)
- Resembles assembler

0001	LD	A	
0002	AND		B
0003	ST	C	
0004			
0005			

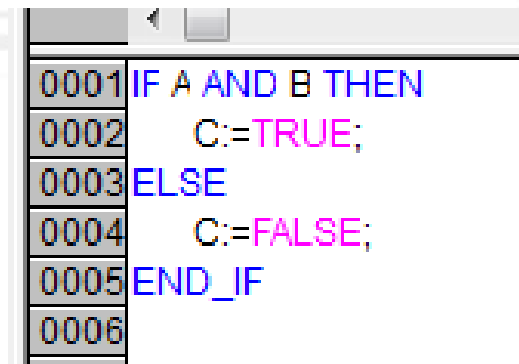
LD - Ladder

- Graphical language
- Network oriented
- Good for programming boolean expressions



ST – Structured Text

- Textual language
- High level language
- Resembles Pascal
- Best language for conditional execution and loops

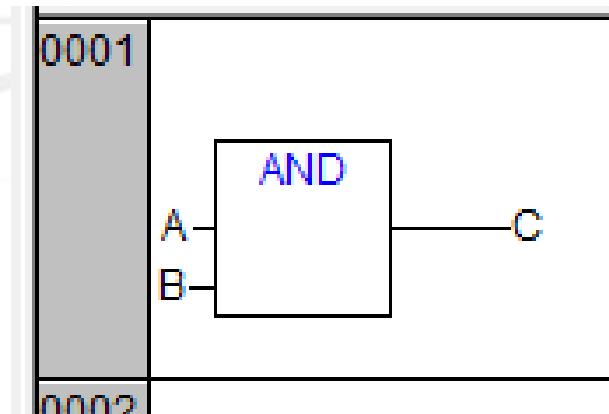


A screenshot of a code editor window displaying Structured Text (ST) code. The code is organized into a table-like structure with line numbers in the first column and code statements in the second column. The statements are: 0001 IF A AND B THEN, 0002 C:=TRUE;, 0003 ELSE, 0004 C:=FALSE;, 0005 END_IF, and 0006. The code is color-coded: 'IF', 'AND', 'THEN', 'ELSE', and 'END_IF' are in blue, while 'C:=TRUE;' and 'C:=FALSE;' are in magenta.

0001	IF A AND B THEN
0002	C:=TRUE;
0003	ELSE
0004	C:=FALSE;
0005	END_IF
0006	

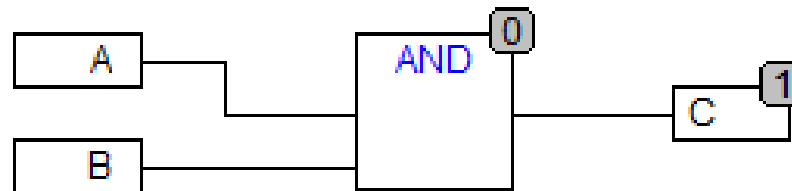
FBD – Function Block Diagram

- Graphical language
- Network oriented
- Easy to understand



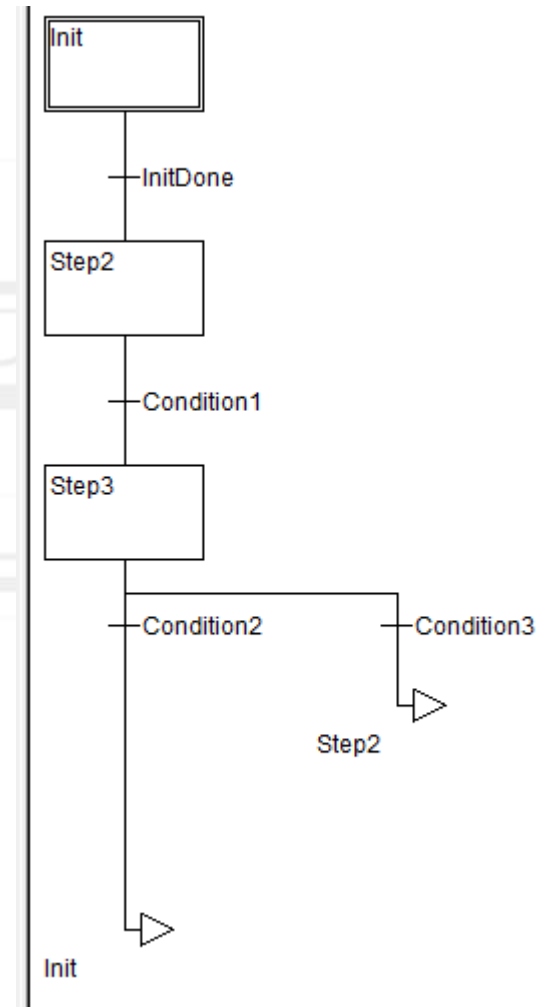
CFC – Continuous Function Chart

- Close to FBD
- Blocks can be placed freely
- Possible to loop outputs back to inputs
- Not defined in IEC 61131-3



SFC – Sequential Flow Chart

- Graphical language
- Consists of steps and transitions
- Good for creating sequences
- Each step can be programmed with any language



- Function Blocks and Programs can be programmed with any programming language
- Functions can be programmed with IL, LD, ST, FBD and CFC, but not with SFC

Data Types

- Structures containing basic variable types
- Resembles structures in C language

```
0001 TYPE ExampleStructure :  
0002 STRUCT  
0003     Variable1:BOOL;  
0004     Variable2:INT;  
0005     Variable3:INT;  
0006     Variable4:STRING[100];  
0007 END_STRUCT  
0008 END_TYPE  
0009
```

Visualizations

- Graphical tool to design user interface
- Easy to connect POU variables or CAN values to UI
- Can not be used with Epec I/O modules, only 2040 display
- More of visualization in another presentation

Resources

- Global variables
- Libraries
- PLC Browser
 - I/O configuration
- Sampling Trace
- Target Settings
- Tasks

Variable types

Variable type	Size (bits)	Low limit	High limit
BOOL	1	FALSE (0)	TRUE (1)
BYTE	8	0	255
USINT	8	0	255
SINT	8	-128	127
WORD	16	0	65535
UINT	16	0	65535
INT	16	-32768	32768
DWORD	32	0	4294967295
UDINT	32	0	4294967295
DINT	32	-2147483648	2147483647
REAL	32		

Variable types

Variable type	Size (bits)	Low limit	High limit
ULINT	64	0	$2^{64}-1$
LINT	64	-2^{63}	$2^{63}-1$
LREAL	64		
STRING	(default 80)		
TIME			
POINTER			
ARRAY			

Declaring a variable

- Manually
- Automatic declaration
- Local or global
- Input, output or normal variable
- Initial value

Manually declaring a variable

- Declaration area of a POU

`<name>: <type> (:=<initial value>);`

- Example of declaring a normal variable:

`VAR`

`Variable1: BOOL:=FALSE;`

`END_VAR`

- Input variables are declared between `VAR_INPUT/END_VAR` tags
- Output variables between `VAR_OUTPUT/END_VAR` tags

Automatic declaration of a variable

- Write a new variable name in code editor
- Automatic declaration window pops up
- Define variable type, class and initial value

Declare Variable

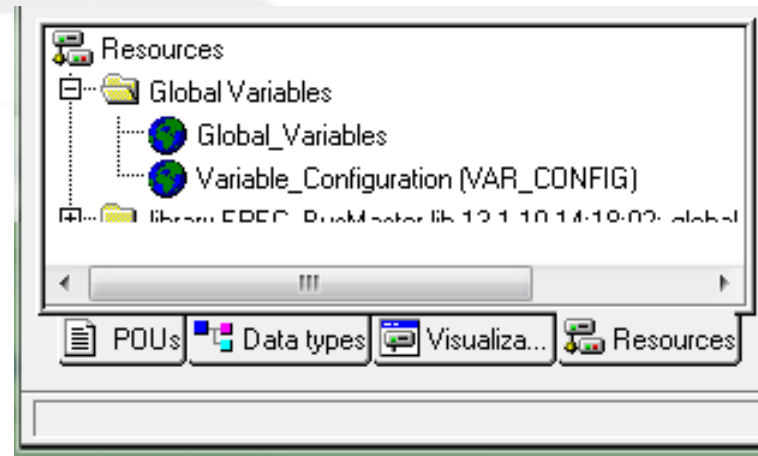
<u>C</u> lass VAR	<u>N</u> ame Variable1	<u>T</u> ype BOOL
<u>S</u> ymbol list _IO	<u>I</u> nitial Value	<u>A</u> ddress
<u>C</u> omment:		

☐ CONSTANT
☐ RETAIN

OK
Cancel

Global variables

- Global variables are declared in Resources tab
- Declare global variables either manually, or select Class VAR_GLOBAL in declare variable window



Variable and POU names

- Letters and numbers
- Not case sensitive
- Must start with a letter
- Only single underscore
- No spaces
- Examples
 - Variable1, VARIABLE1, Variable_1

Operators

- Boolean operators
 - AND, OR, XOR
- Analog operators
 - ADD, SUB, MUL, DIV, MOD
- Compare operators
 - EQ, LT, GT, LE, GE
- Shift and rotate operators
 - SHL, SHR, ROL, ROR

Operators

- Real operators
 - ABS, TRUNC, EXPT, SQRT
- Logarithmic operators
 - EXP, LN, LOG
- Trigonometric operators
 - SIN, COS, TAN, ASIN, ACOS, ATAN
- Select operators
 - MIN, MAX, SEL, MUX

Libraries

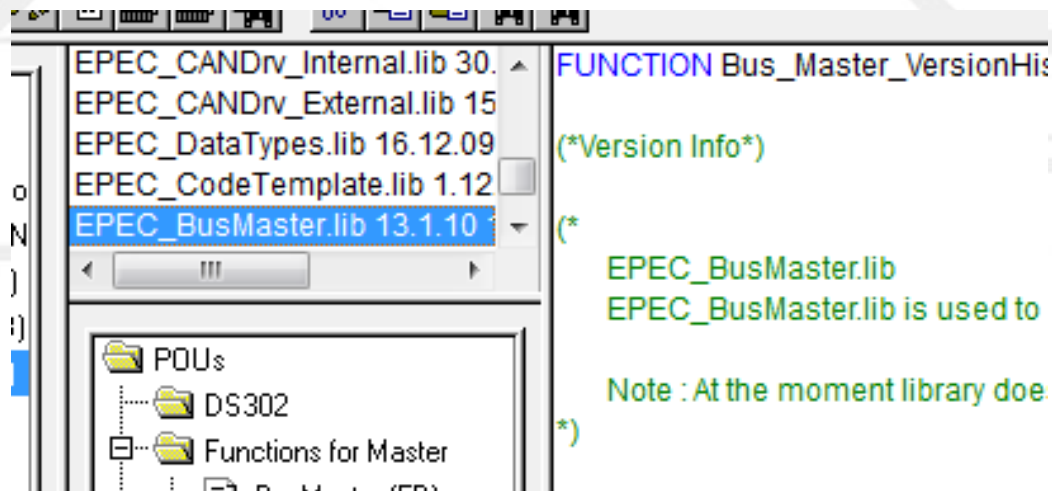
- External files containing pre-defined POU's and data types
- Can be easily built with CoDeSys
- Same libraries can be used with I/O module and 2040 display

Standard library

- String functions
 - LEN, LEFT, RIGHT, MID, CONCAT, INSERT, DELETE, REPLACE, FIND
- Edge detection
 - F_TRIG, R_TRIG
- Counter
 - CTU, CTD, CTUD
- Timers
 - TP, TON, TOF

Libraries

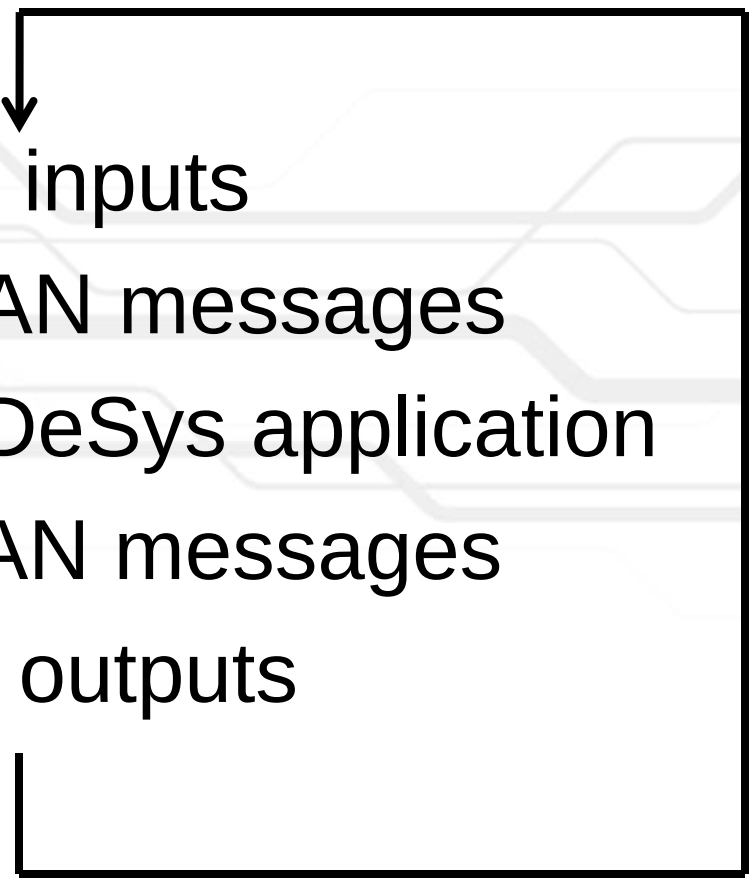
- Additional libraries can be added with library manager
- Epec libraries are installed with target



Architecture



Program cycle of a CoDeSys Application

- 
- Read all inputs
 - Read CAN messages
 - Run CoDeSys application
 - Write CAN messages
 - Write all outputs