

CODESYS Basics

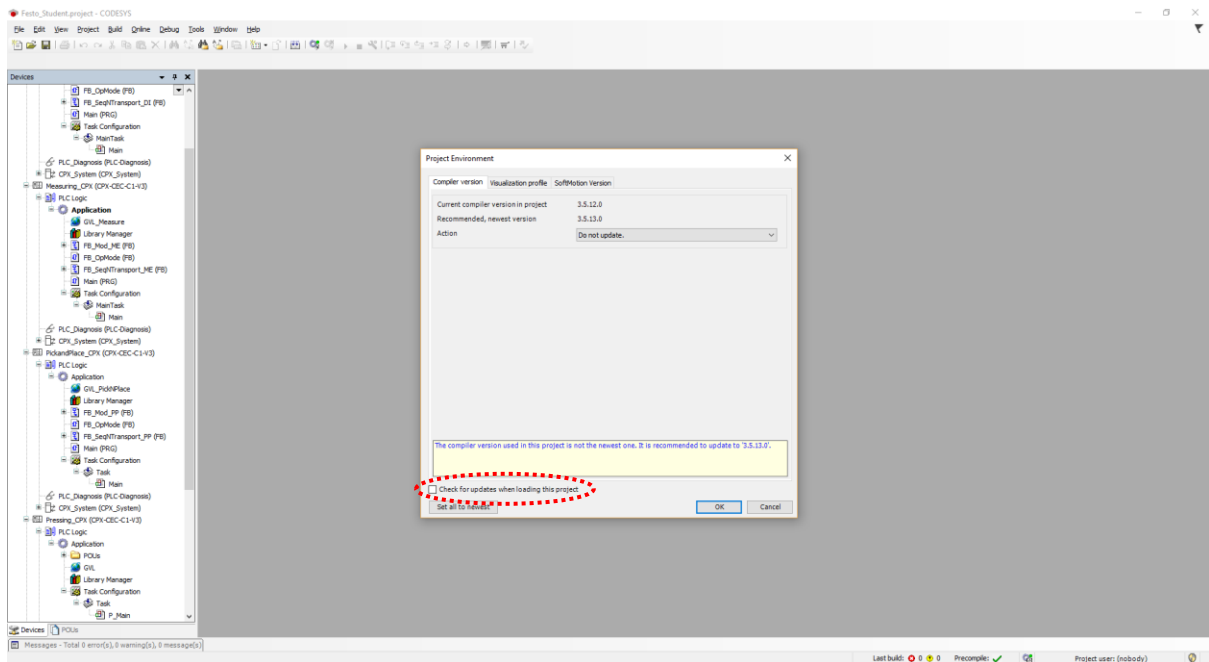
GENG5508: Robotics

Project – Festo

Getting started

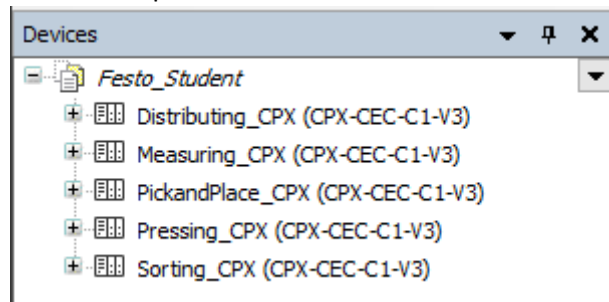
Form groups of 3-4 people. Only one computer needs to have CODESYS installed per group.

1. Download 'CODESYS3.5.13.0.exe' and 'Festo_Student.project' from <https://roblab.org/courses/robman/labs/info-Festo/>
2. Double-click CODESYS3.5.13.0.exe. Follow the installation wizard. REQUIRES WINDOWS 7 OR HIGHER – NOT COMPATIBLE WITH MAC.
3. Start CODESYS V3.5 SP13 and open 'Festo_Student.project'
4. In 'Project Environment', untick 'Check for updates when loading the project' and click 'OK'.

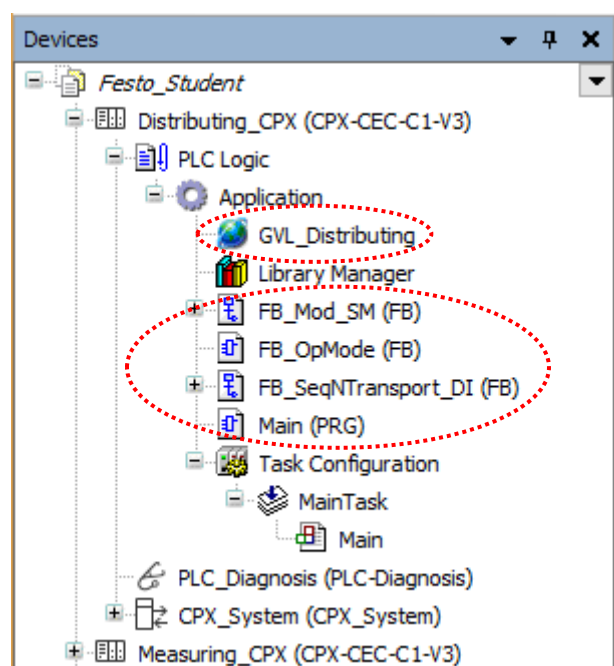


CODESYS Familiarisation

- There are five modules in the Festo Modular Production System (Festo MPS):
 - Distribution
 - Measurement
 - Pick-and-place
 - Fluid press
 - Sorting
- Each module in the Festo MPS is associated with a Programmable Logic Controller (PLC) shown in the left-hand 'Devices' pane.



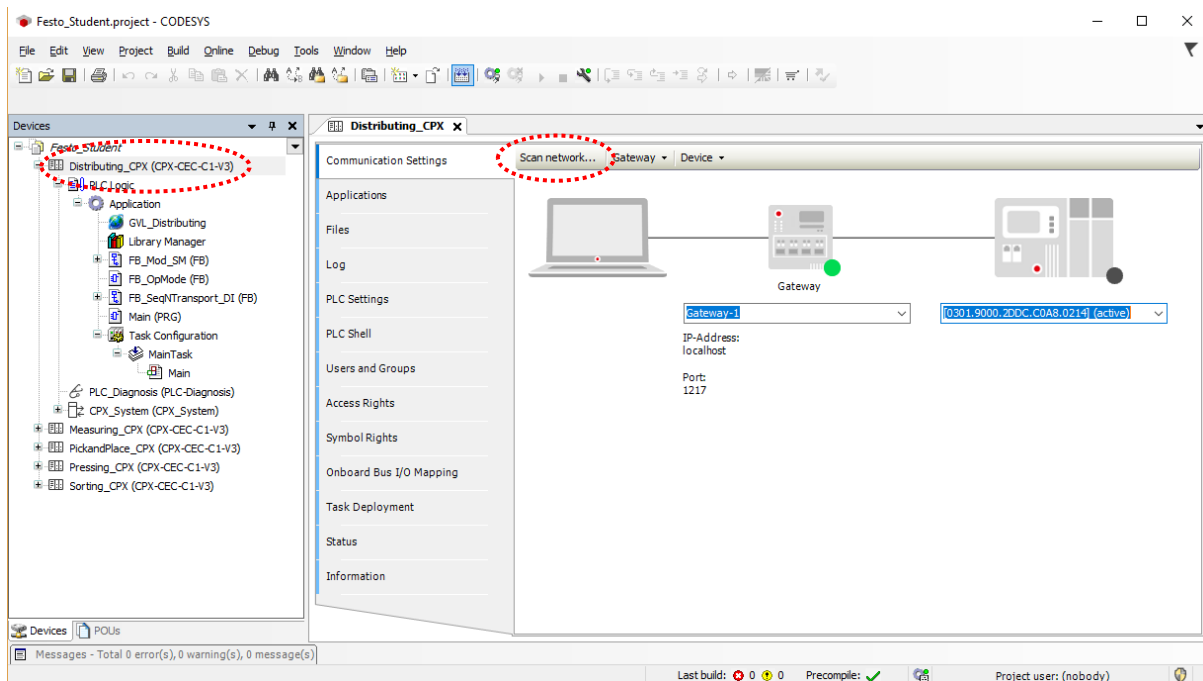
- The code to program each module is contained in several files. The 'GVL_<module_name>' file is the Global Variable List file, which declares global variables accessible by all scripts for a single module.



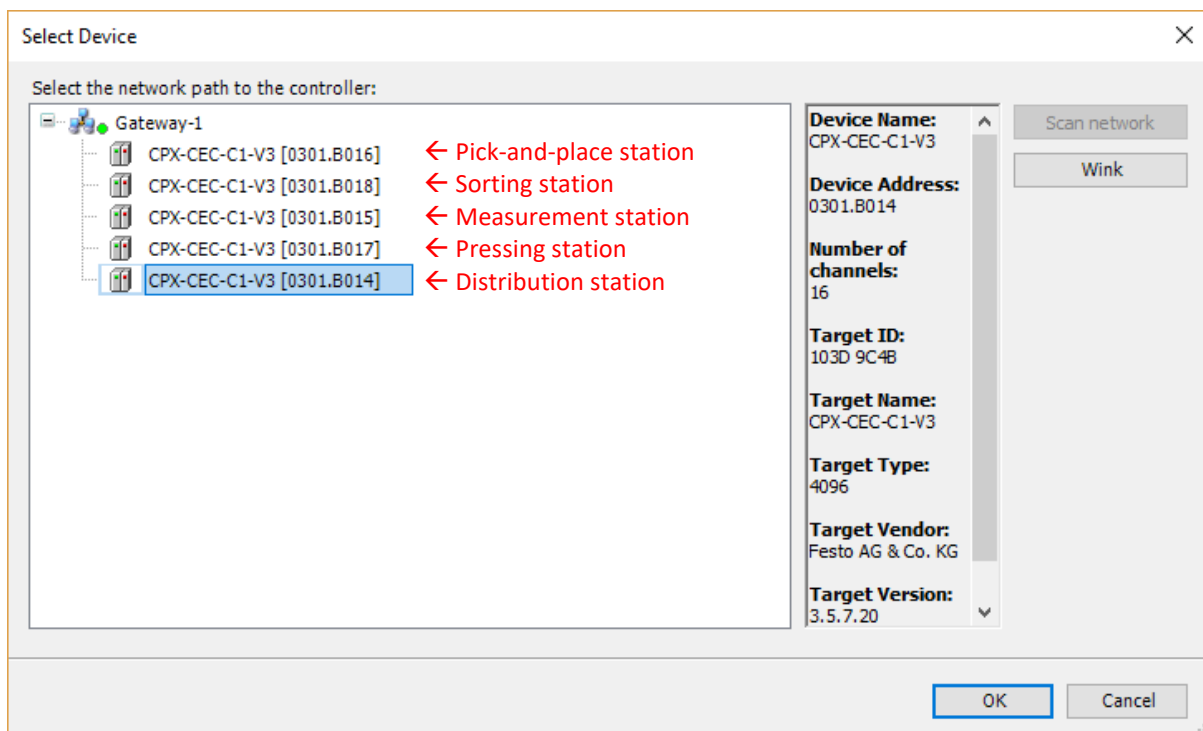
Uploading to a module

The following steps will describe how to log into the PLC for a given module and upload code:

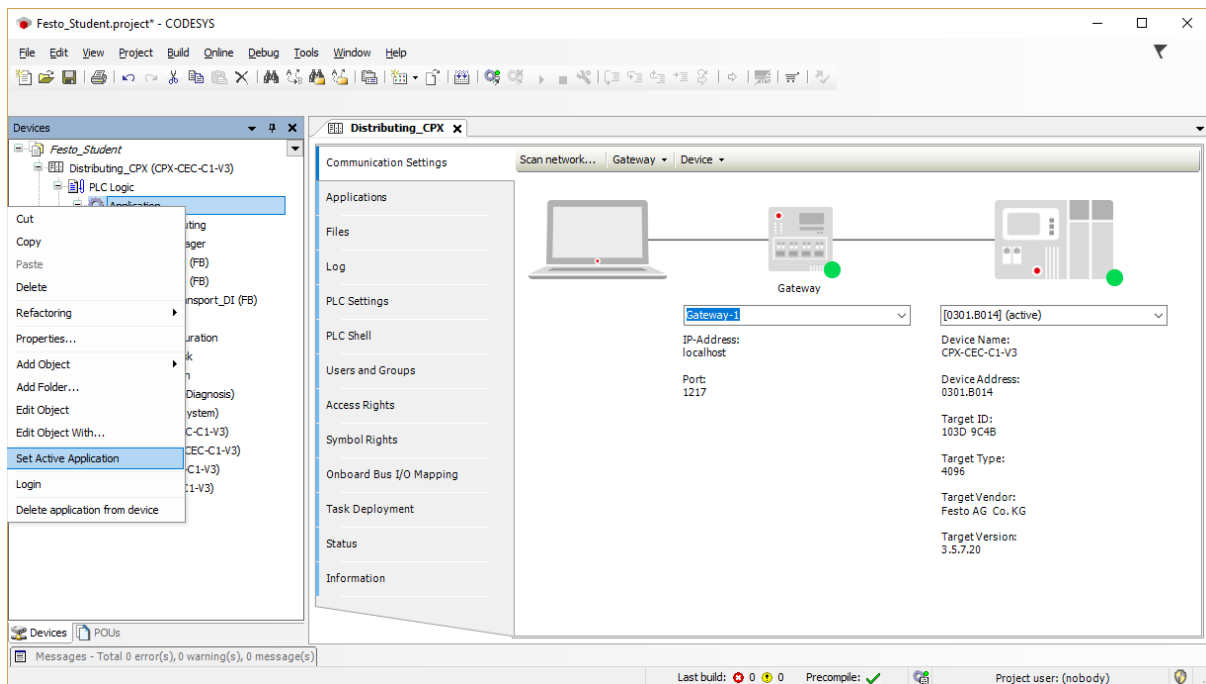
1. Before modifying any programs on the Festo MPS, ensure the rotary switch is NOT set to 0 or 1 (e.g., it should be set to 2). The lab demonstrator will show how to do this.
2. Connect to the 'Festo' router (password 'festo2010').
3. Double-click on the module name to bring up the 'Communication Settings' window, then click 'Scan Network'.



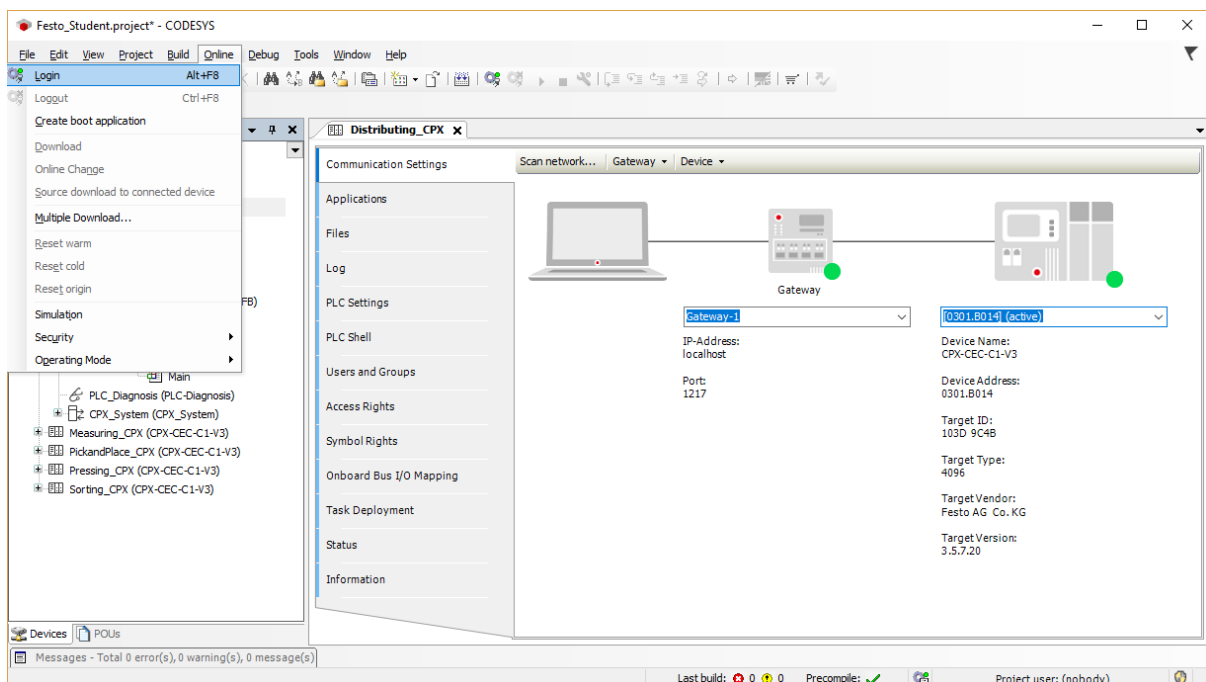
4. Select the PLC to connect to:



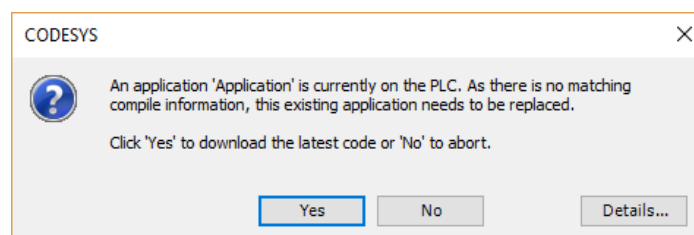
- Under the selected PLC, right-click 'Application' and click 'Set Active Application'. The selected application will become **bold**.



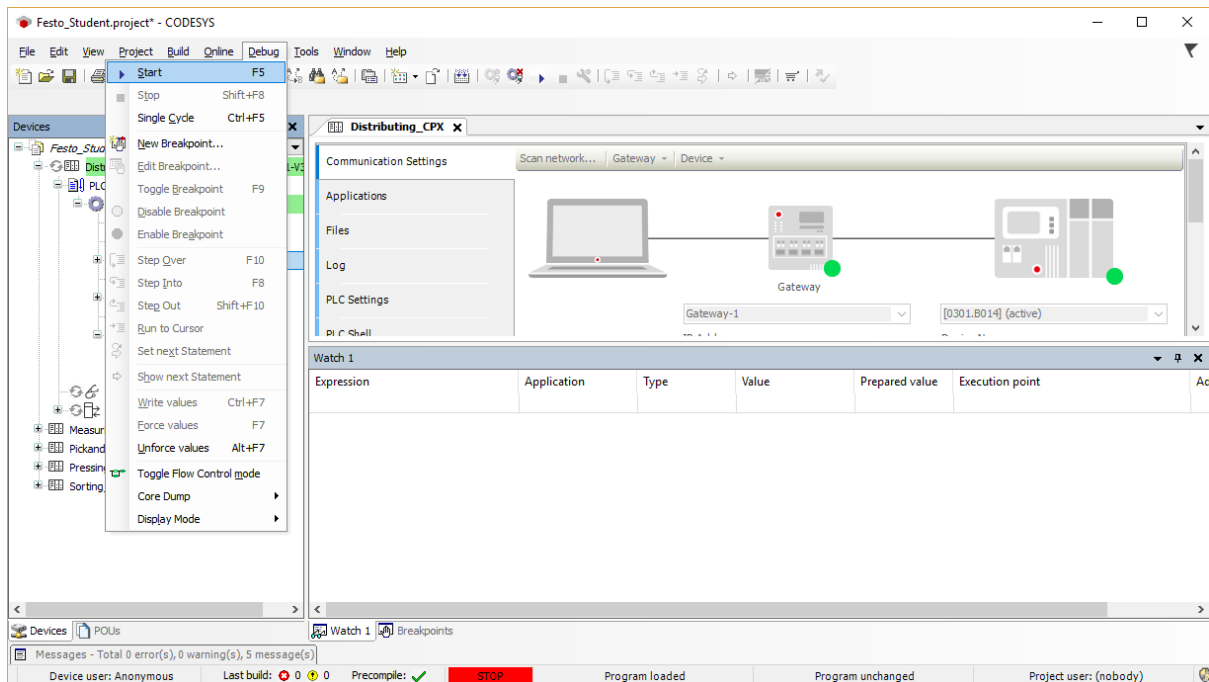
- Login to the PLC by clicking 'Online' -> 'Login'



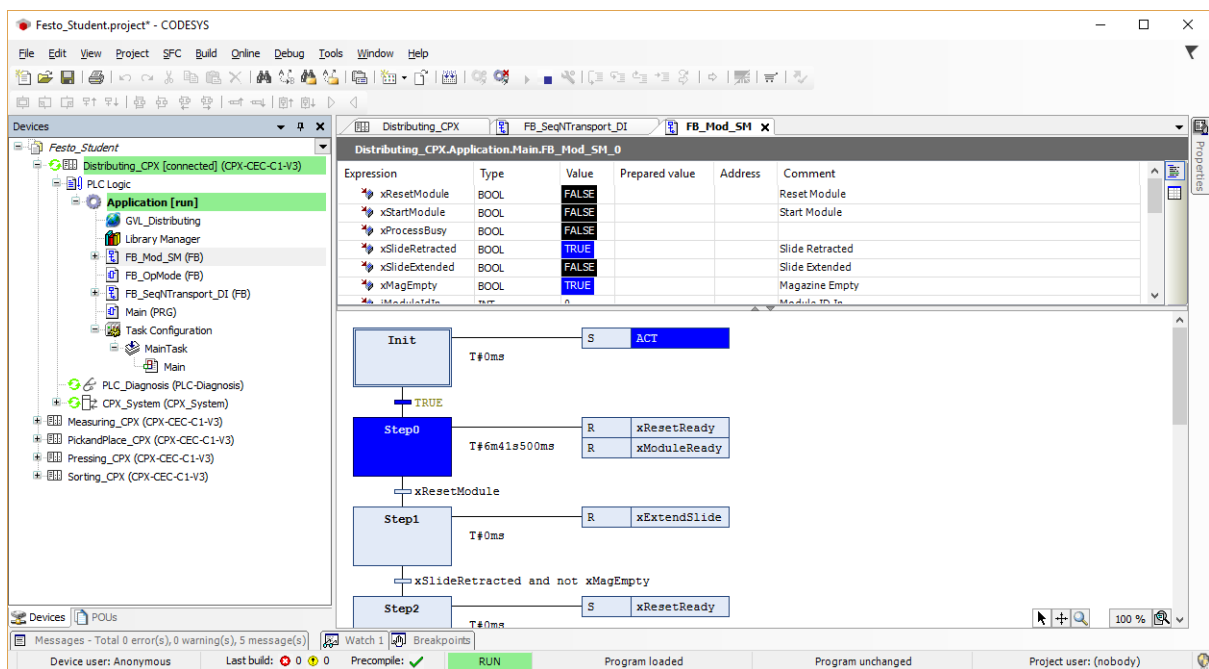
- Click 'Yes' at the prompt to upload your code to the PLC.



8. Run the uploaded code by clicking 'Debug' -> 'Start'. 'Run' will appear next to 'Application' in the Devices menu. You can now interact with the control panel on the module to start the production line.



9. Optional: Open a script to monitor the current progress through the code.



Important scripts

The following scripts will be particularly useful for Lab 1.

Main

The 'Main' script shows the mapping between global variables and local variables within Programming Organisational Units (POUs).

The screenshot displays the CODESYS IDE interface for the 'Main' script. At the top, a table lists global variables:

Expression	Type	Value	Prepared value	Address	Comment
FB_OpMode_0	FB_OpMode				
xKeySwitch	BOOL	FALSE			Schlüsselschalter Auto-Manuell / Key Switch auto-manual
xStart	BOOL	FALSE			Taster Start / Start button
xStop	BOOL	TRUE			Taster Stop (Offner) / Stop Button (normally dosed)
xReset	BOOL	FALSE			Taster Richten / Reset button
xEMG_Stop	BOOL	TRUE			Not Aus Schlagtaster / Emergency Stop
xI4	BOOL	FALSE			Bedienfeld Eingang 4 / Panel Input 4
xI5	BOOL	FALSE			Bedienfeld Eingang 5 / Panel Input 5

Below this table, the variable declaration section of the 'Main' POU is shown, including:

- Global variables:** xKeySwitch, xStart, xStop, xReset, xEMG_Stop, xI4, xI5, xErrorSignal1, xResetReadyMM, xResetReadyModule1, AlwaysTRUE, xProcessBusyMM, xNextStationFreeMM, xConvForward, xConvBackward, xExtendSeparator, xStartModule1, xProcessBusy, xResetReady, xInitModules, xEnAutoMM, xNextStationFreeMM, xInitModules.
- POU inputs:** xKeySwitch, xStart, xStop, xReset, xEMG_Stop, xI4, xI5, xErrorSignal1, xResetReadyMM, xResetReadyModule1, AlwaysTRUE, xProcessBusyMM, xNextStationFreeMM, xConvForward, xConvBackward, xExtendSeparator, xStartModule1, xProcessBusy, xResetReady, xInitModules, xEnAutoMM, xNextStationFreeMM, xInitModules.
- POU outputs:** xLEDStart, xLEDReset, xLED1, xLED2, xD, xStatusGreen, xStatusYellow, xStatusRed, xResetModules, xDeactivateModules, xEnAutoMM, xNextStationFreeMM, xInitModules, G1BF1_A2, G1MB1, xStartModuleSM, xProcessBusyMM, xResetReadyMM, vBeetReady.

The ladder logic diagram shows the implementation of these variables. Red dashed circles highlight the global variable declarations and the POU input/output connections. The status bar at the bottom indicates the program is in a 'STOP' state.

GVL_<module_name>

The 'GVL_<module_name>' file shows the mapping between global variables and the PLC input/output address bits. It also shows the current state (true/false) of the variables.

The screenshot displays the CODESYS interface for a project named 'Festo_Student.project'. The main window shows a table titled 'Distributing_CPX.Application.GVL_Distributing' with the following columns: Expression, Type, Value, Prepared value, Address, and Comment. A red dashed oval highlights the 'Expression' and 'Value' columns, with a red arrow pointing to the text 'Global variables' on the left. Another red dashed oval highlights the 'Address' column, with a red arrow pointing to the text 'PLC address bit' on the right. A third red dashed oval highlights the 'Value' column, with a red arrow pointing to the text 'Variable state' at the bottom. The table lists various global variables such as G1B, C2B, SF, XD, G1BFL, G1MB, C2MB, PF, XD, indLight, xReset, xEnAuto, xDeactivate, xProcessBusy, xResetReady, xNextStation, xReadyMod, xStartModule, xInitModule, Always, ErrorSignal, and xErrorSignal, along with their current states and PLC addresses.

Expression	Type	Value	Prepared value	Address	Comment
G1B G1	BIT	FALSE		%IX1.0	Werkstück bei Bandanfang / workpiece AT conveyor start
G1B G2	BIT	FALSE		%IX1.1	Werkstück bei Bandmitte / workpiece AT conveyor center
G1B G3	BIT	FALSE		%IX1.2	Werkstück bei Bandende / workpiece AT conveyor end
C2B G1	BIT	TRUE		%IX1.4	Schieber eingefahren / slide retracted
C2B G2	BIT	FALSE		%IX1.5	Schieber ausgefahren / slide extended
C2B G3	BIT	TRUE		%IX1.6	Magazin leer / magazine empty
SF1	BIT	FALSE		%IX2.0	Taster Start / Start button
SF2	BIT	TRUE		%IX2.1	Taster Stop (Öffner) / Stop Button (normally closed)
SF4	BIT	FALSE		%IX2.2	Schlüsselschalter Auto-Manuell / Key Switch auto-manual
SF3	BIT	FALSE		%IX2.3	Taster Richten / Reset button
SF5	BIT	TRUE		%IX2.5	Not Aus Schlagtaster / Emergency Stop
XD6	BIT	FALSE		%IX2.6	Folgestation belegt/ dwonstreamstation busy
XD7	BIT	FALSE		%IX2.7	Folgestation belegt/ dwonstreamstation busy
G1BFL_A1	BIT	FALSE		%QX0.0	Bandmotor vorwärts / Conveyor motor forward
G1BFL_A2	BIT	FALSE		%QX0.1	Bandmotor rückwärts / Conveyor motor backwards
G1MB1	BIT	FALSE		%QX0.2	Vereinzelr ausfahren / Extend separator
C2MB1	BIT	FALSE		%QX0.4	Schieber ausfahren / Extend slide
PF1	BIT	FALSE		%QX1.0	Leuchtmelder Start / Start indicator light
PF2	BIT	FALSE		%QX1.1	Leuchtmelder Reset / Reset indicator light
PF3	BIT	FALSE		%QX1.2	Leuchte Q1 / Indicator light Q1
PF4	BIT	FALSE		%QX1.3	Leuchte Q2 / Indicator light Q2
XD4	BIT	FALSE		%QX1.4	Station frei / Station free
indLightGn	BIT	FALSE		%QX1.5	Ampel grün / indicator lights green
indLightYe	BIT	FALSE		%QX1.6	Ampel gelb / indicator lights yellow
indLightRd	BIT	FALSE		%QX1.7	Ampel rot / indicator lights red
xResetModules	BOOL	FALSE			Alle Module resettet / Reset all Moduls
xResetReadyMod1	BOOL	FALSE			Module 1 ist in reset Position / Modul 1 is in reset position
xResetReadyMod2	BOOL	FALSE			Module 2 ist in reset Position / Modul 2 is in reset position
xResetReadyMod3	BOOL	FALSE			Module 3 ist in reset Position / Modul 3 is in reset position
xEnAutoMM	BOOL	FALSE			Automatikmodus aktiv / Automode activated
xDeactivateModu...	BOOL	TRUE			
xProcessBusyMM	BOOL	FALSE			Station arbeitet / Station is busy
xResetReadyMM	BOOL	FALSE			Modul SeqTransport ist fertig / Modul SeqTransport is ready
xNextStationFree...	BOOL	FALSE			Nächste Station ist frei / Next Station is free
xReadyModSV	BOOL	FALSE			Modul ist fertig / Modul is ready
xStartModuleSV	BOOL	FALSE			Modul starten / Start modul
xInitModuleSV	BOOL	FALSE			Moduel auf Inischnitt setzen / Set moduls to init step
AlwaysTRUE	BOOL	TRUE			Immer true / Always true
AlwaysFALSE	BOOL	FALSE			Immer false / Always false
ErrorSignal1	BOOL	TRUE			Error Signal 1 / Error signal 1
ErrorSignal2	BOOL	FALSE			Error Signal 2 / Error signal 2
xErrorSignal3	BOOL	FALSE			Error Signal 3 / Error signal 3