

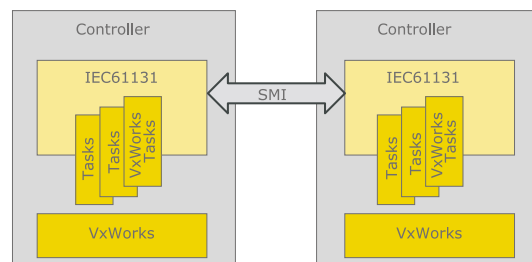
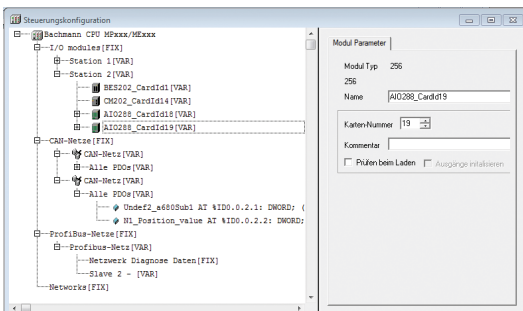
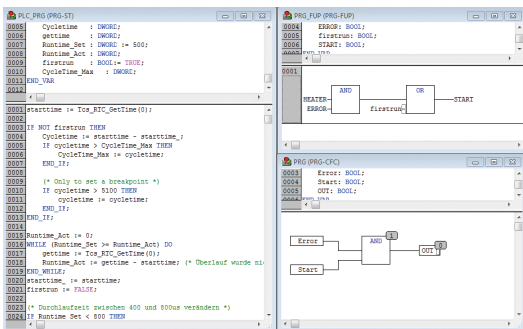
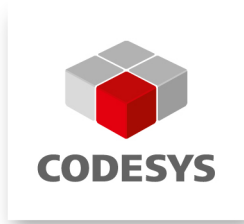
M-PLC Programming interface

PLC programming interface acc. to IEC 61131-3

M-PLC is a complete development environment for programming the M200 controller in accordance with IEC 61131-3. The M-PLC puts a simple approach to the powerful IEC language at the disposal of the PLC programmer. Use of the edit and debugging functions is based on the proven development program environments of advanced programming languages.

Features

- Easy entry into IEC 61131-3
- All languages defined in the IEC 61131-3 standard are supported
- Editor and debugging functions
- Provided libraries with pre-finished and tested functions



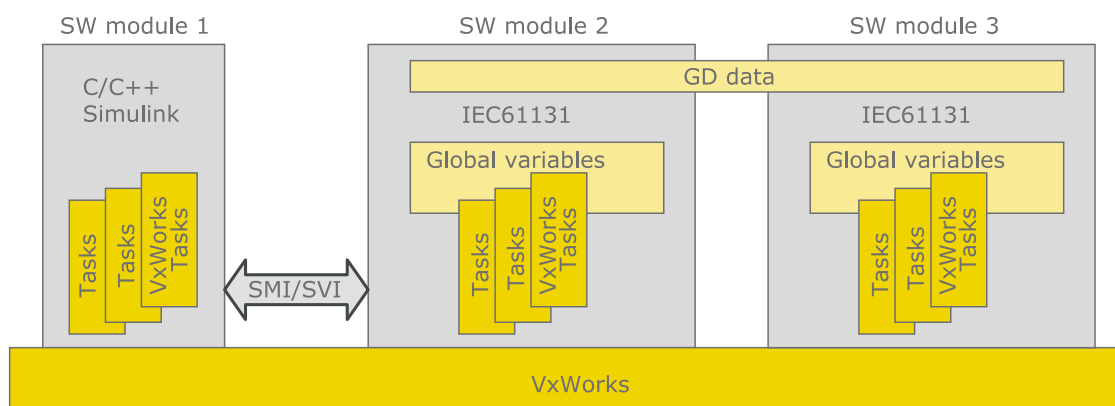
Programming

- Editors for programming in all IEC 61131-3 languages:
 - IL - Instruction List
 - ST - Structured Text
 - LD - Ladder Diagram
 - FBD - Function Block Diagram
 - SFC - Sequential Function Chart
 - CFC - Continuous Function Chart
- Syntax highlighting for key words defined in IEC 61131-3
- Graphic project navigation bar (based on Windows Explorer)
- Support of all elementary IEC 61131-3 data types to 64 bit
- Support of fields, structures and pointers
- Graphic controller and task configuration
- CAN objects from *.dcf files in M-PLC can be applied as symbols
- Profibus objects from *.2bf files in M-PLC can be applied as symbols
- Library management for creation and management of libraries
- Watch manager and recipe manager (freely definable variable monitoring)
- M-PLC can be started several times (multiple projects open and online in parallel)
- Automatic project backup and creation of data backup
- Project archiving on the controller
- Password protection for projects
- Offline simulation
- Data exchange system-wide via SVI/SMI interface programming
- Online help

Runtime system

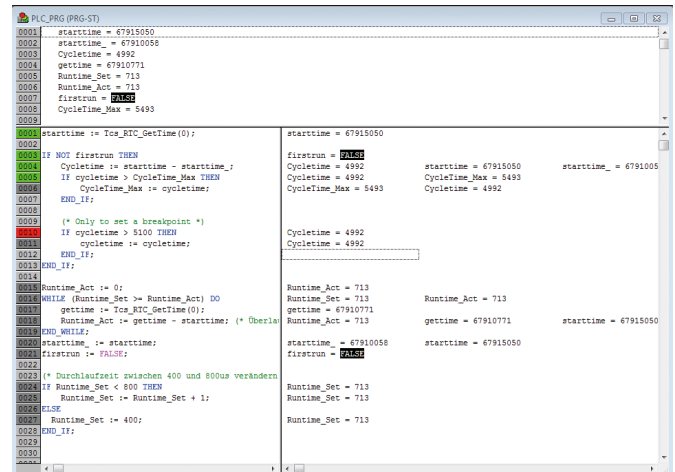
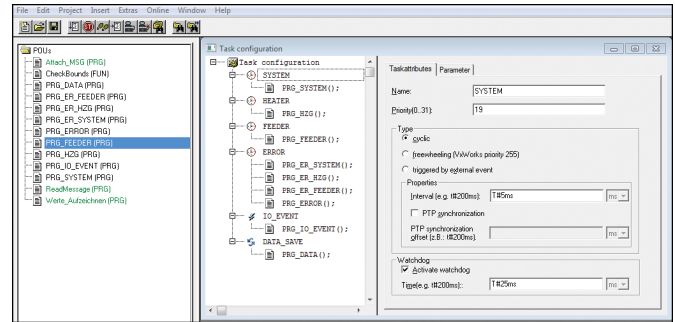
- Any number of running M-PLC projects on one processor¹⁾
- Every PLC project supports up to 16 separate tasks
- Each task can be configured independently of the others in terms of priority, task call-up mode [event, timetrigger, free running, cyclic, sync]
- Actual multitasking via operating system tasks (VxWorks)
- Communication between software modules (pre-compiled units)
- Between different tasks of an IEC 61131 module: global variables or SMI/SVI
- Between IEC 61131 modules: GD marker range or SMI/SVI
- Between IEC 61131 modules and other modules (C/C++, Simulink): SMI/SVI
- Between any modules on different controllers: SMI
- Data link to visualization/SCADA/BDE
- Standard protocols: OPC DA, IEC 61850 / IEC 61400, Modbus etc.
- Libraries for integration into separate software (for C/C++, C#.NET, Java)
- Support of non-volatile buffered data (RETAIN marker)

¹⁾ Maximum 256 user software modules



Online functions

- Online change (exchange of blocks in running operation) also for multitasking projects
- Monitoring of all project variables
- Writing and forcing of variables
- Single cycle, single step and breakpoints
- Sequence control (program lines running through will be displayed)
- Recording and graphic presentation of project variables – trace



Manufacturer libraries

STANDARD	IEC 61131-3 Standard functions and function blocks
SMI_PLC	Functions for fast communication between modules and different controllers (SMI/SVI)
CONT_PLC	7 standard PID controllers
MIO_PLC	Functions for direct access to hardware I/Os
EHD_PLC	Functions for entering and managing errors in the EHD
UTIL_PLC	Functions for access to RTC, runtime system info, special conversions

FILE_PLC	Functions for access to files, directories and serial interfaces
CIA405	Functions for access to CAN
DN_PLC	Functions for access to DeviceNet
PB_PLC	Functions for access to PROFIBUS
USS_PLC	Functions for communication with „Micromaster“ frequency converters

Extensions relative to CoDeSys

- Runtime-optimized process image (only channels being used are processed)
- All I/O channels can be displayed with symbolic names in the process image
- Addressing for inputs/outputs: undelayed access that bypasses the process image
- Setting of the interval time and watchdog time in the controller configuration
- Multiple, concurrently running PLC projects on one CPU with different priorities / interval times
- Actual operating system tasks within the PLC projects
- GD marker range for shared data from multiple projects on one CPU
- RD marker range for remanent (non-volatile buffered) data
- Optionally, the normal marker range (MX to MR) can also be placed in the non-volatile buffered memory range
- Interface in the runtime system for libraries
- Initialization (C), deinitialization, version check, memory management, management of background tasks, provision for online change
- Support for exception handling in the runtime system
- Display of the status „in error handling“ (ERROR) in the user interface
- Implementation of the I/O modules of the process image in the runtime system
- Optimized for performance, flexibility and combinability, shared use of an I/O module by multiple projects / SW modules possible
- Saving/loading of the entire project including sources to/from the controller
- Transparency of the global variables (markers, symbolic variables, structures) via the SVI/SMI interface control and visualization-wide

Integration in the SolutionCenter (projects are SW modules)

Generation of an executable *.m file with configuration information

- Task interval time measurement can be queried in the project itself and in Device Manager
- Watchdog integration
- Execution possible in an application layer (memory protection)
- Install, start, stop, reset and delete of PLC projects in Device Manager
- The state machine of a PLC project in the runtime system always corresponds to the model defined for SW modules
- Integration via M200 system debug mode and M200 system logbook
- Start of a project when booting through entry in mconfig.ini