

M-PLC Programmieroberfläche

SPS-Programmieroberfläche nach IEC 61131-3

M-PLC ist eine komplette Entwicklungsumgebung für die Programmierung der M1-Steuerung nach IEC 61131-3. Das M-PLC ermöglicht dem SPS-Programmierer einen einfachen Einstieg in die mächtigen Sprachmittel der IEC. Die Benutzung der Editoren und der Debugging-Funktionen hat die ausgereiften Entwicklungsumgebungen höherer Programmiersprachen zum Vorbild.

- Einfacher Einstieg in die IEC 61131-3
- Alle in der Norm IEC 61131-3 definierten Sprachen werden unterstützt.
- Editor und Debugging-Funktionen
- Mitgelieferte Bibliotheken mit vorgefertigten und getesteten Funktionen

Programmierung

- Editoren zur Programmierung in allen IEC61131-3 Sprachen
 - AWL Anweisungsliste
 - ST Strukturierter Text
 - KOP Kontaktplan
 - FUP Funktionsplan
 - AS Ablaufsprache
 - CFC Freigrafischer Funktionsplan

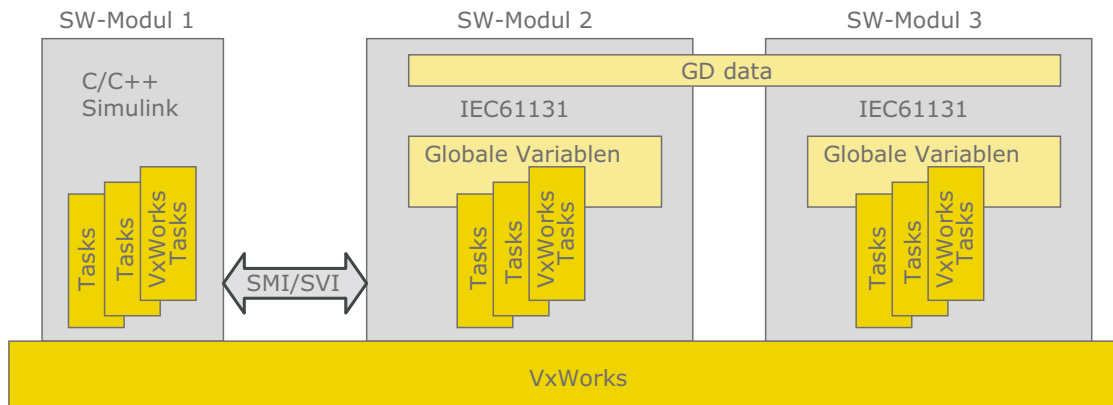
- Syntax-Highlighting (Hervorhebung) für in der IEC 61131-3 definierte Schlüsselwörter
- Grafische Projektnavigationsleiste (angelehnt an Windows Explorer)
- Unterstützung aller elementaren IEC 61131-3 Datentypen bis 64 Bit
- Unterstützung von Feldern, Strukturen und Pointern
- Grafische Steuerungs- und Taskkonfiguration
- CAN-Objekte aus *.dcf-Dateien in M-PLC als Symbole übernehmbar
- Profibus-Objekte aus *.2bf-Dateien in M-PLC als Symbole übernehmbar
- Bibliotheksverwaltung zum Anlegen und Verwalten von Bibliotheken
- Watch- und Rezepturverwalter (frei definierbares Variablenmonitoring)
- M-PLC kann mehrfach gestartet werden (mehrere Projekte parallel geöffnet und online)
- Automatische Projektsicherung und Erstellung von Backup
- Projektarchivierung auf der Steuerung
- Passwortschutz für Projekte
- Offline-Simulation
- Datenaustausch systemweit über SVI/SMI-Schnittstellen-Programmierung
- Online-Hilfe

Laufzeitsystem

- Beliebige viele laufende M-PLC Projekte auf einem Prozessor ¹⁾
- Jedes PLC-Projekt unterstützt bis zu 16 eigene Tasks
- Jeder Task kann unabhängig von den anderen konfiguriert werden in Bezug auf (Priorität, Task-Aufrufmodus [Event, Timetrigger, Free running, zyklisch, Sync])
- Echtes Multitasking durch Betriebssystemtasks (VxWorks)
- Kommunikation zwischen Software-Modulen (vorkompilierten Einheiten)
- Zwischen verschiedenen Tasks eines IEC 61131 Moduls: Globale Variablen oder SMI/SVI
- Zwischen IEC 61131-Modulen: GD-Merker-

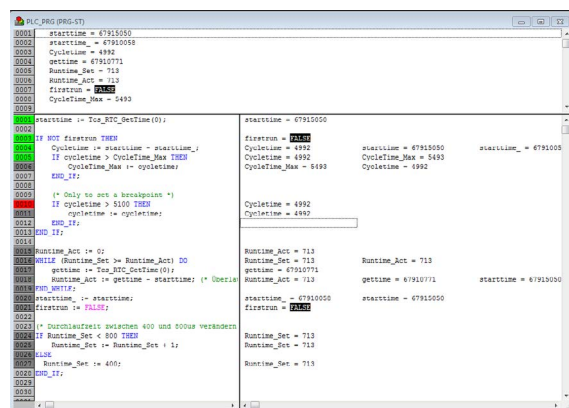
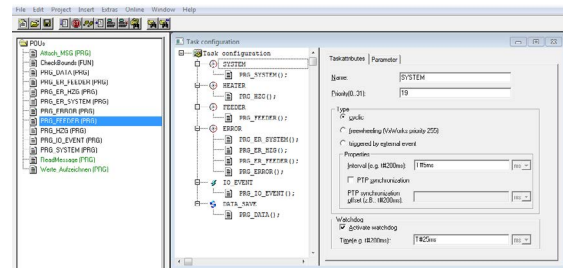
- Bereich oder SMI/SVI
- Zwischen IEC 61131-Modulen und anderen Modulen (C/C++, Simulink): SMI/SVI
- Zwischen beliebigen Modulen auf unterschiedlichen Steuerungen: SMI
- Datenanbindung an Visualisierung/SCADA/BDE
- Standardprotokolle: OPC DA, IEC 61850/ IEC 61400, Modbus usw.
- Bibliotheken zur Einbindung in eigene Software (für C/C++, C#.NET, Java)
- Unterstützung von permanent gepufferten Daten (RETAIN-Merker)

1) Maximal 256 User Software-Module



Online-Funktionen

- Online-Change (Austauschen von Bausteinen im laufenden Betrieb) auch bei Multitasking Projekten
- Monitoring sämtlicher Projektvariablen
- Schreiben und Forcen von Variablen
- Einzelzyklus, Einzelschritt und Breakpoints
- Ablaufkontrolle (durchlaufende Programmzeilen werden angezeigt)
- Aufzeichnen und grafisches Darstellen von Projektvariablen – Trace



Herstellerbibliotheken

- **STANDARD** IEC 61131-3 Standard-Funktionen und Funktionsbausteine
- **SMI_PLC** Funktionen für schnelle Kommunikation zwischen Modulen auf unterschiedlichen Steuerungen (SMI/SVI)
- **CONT_PLC** 7 Standard PID-Regler
- **MIO_PLC** Funktionen für den direkten Zugriff auf Hardware-E/As
- **EHD_PLC** Funktionen zum eintragen und Verwalten von Fehlern im EHD
- **UTIL_PLC** Funktionen für Zugriff auf RTC, Laufzeit-systeminfos, spezielle Konvertierungen
- **FILE_PLC** Funktionen für Zugriff auf Dateien, Verzeichnisse und serielle Schnittstellen
- **CIA405** Funktionen für Zugriff auf CAN
- **DN_PLC** Funktionen für Zugriff auf Device-Net
- **PB_PLC** Funktionen für Zugriff auf Profibus
- **USS_PLC** Funktionen für die Kommunikation mit »Micro-Master«-Frequenz- umrichtern

Erweiterungen gegenüber CoDeSys

- Laufzeitoptimiertes Prozessabbild (nur benutzte Kanäle werden bearbeitet)
- Alle E/A-Kanäle können mit symbolischen Namen im Prozessabbild
- Adressierung für Ein-/Ausgänge: unverzögerter Zugriff am Prozessabbild vorbei
- Einstellung der Intervallzeit und Watchdog-Zeit in der Steuerungskonfiguration
- Mehrere gleichzeitig laufende PLC-Projekte auf einer CPU mit unterschiedlichen Prioritäten/ Intervallzeiten
- Echte Betriebssystem-Tasks innerhalb der SPS-Projekte
- GD-Merkerbereich für gemeinsame Daten von mehreren Projekten auf einer CPU
- RD-Merkerbereich für remanente (permanentgepufferte) Daten
- Optional kann auch der normale Merkerbereich (MX bis MR) in den permanent gepufferten Speicherbereich gelegt werden.

- Schnittstelle im Laufzeitsystem für Bibliotheken
- Initialisierung (C), Deinitialisierung, Versions- prüfung, Speicherverwaltung, Verwaltung von Hintergrundtasks, Berücksichtigung von Online-Change
- Unterstützung für Ausnahmebehandlung im Laufzeitsystem
- Anzeige des Zustands in Ausnahmebehandlung«
- (ERROR) in der Oberfläche
- Implementierung der E/A-Module des Prozess- abbildes im Laufzeitsystem
- Optimierte Performance, Flexibilität und Kombinierbarkeit, gemeinsame Nutzung eines E/A-Moduls durch mehrere Projekte/SW-Module möglich
- Speichern/Laden des gesamten Projekts inklusive Sourcen auf/von Steuerung möglich
- Transparenz der globalen Variablen
- (Merker, symbolische Variablen, Strukturen) über die SVI/ SMI-Schnittstelle Steuerungs- und visualisierungsweit

Integration in das SolutionCenter (Projekte sind SW-Module)

- Erzeugung einer ausführbaren *.m-Datei mit Konfigurationsinformation
- Task-Intervallzeitmessung, abfragbar im Projekt selbst und Device Manager
 - Watchdog-Anbindung
 - Ausführung in einer Applikationsschicht möglich (Speicherschutz)
 - Install, Start, Stop, Reset und Delete von PLC-Projekten im Device Manager
 - Die Zustandsmaschine eines PLC-Projekts im Laufzeitsystem entspricht dem generell für SW-Module definierten Modell.
 - Integration mittels M1-System-Debugmode und M1-System-Logbuch
 - Start eines Projekts beim Booten durch Eintrag in mconfig.ini