

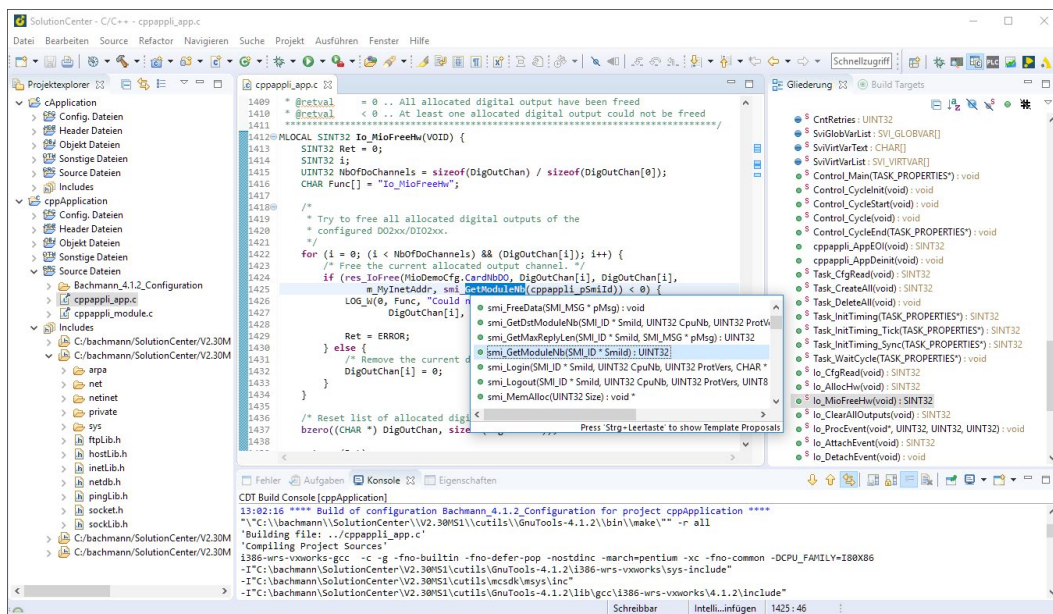
Programmierung



C/C++ Developer

Heutige Automatisierungen lösen komplexe Problemstellungen, die weit über die reine Steuerung der Anlage hinausgehen. Für diese vielschichtigen Aufgabenstellungen ist die Hochsprachen-Programmierung in C/C++ gut geeignet. Diese Sprachen bieten die Möglichkeit einer effizienten, prozeduralen und objektorientierten Programmierung. Mit dem C/C++ Developer wird ein Tool zur Verfügung gestellt, das den Anlagen-Programmierer in allen Phasen der Software-Entwicklung umfassend unterstützt.

Im Kontext der Automatisierung werden Software-Projekte im Zusammenhang mit einer Maschine erstellt. Das zentrale Werkzeug für die Lösung der Automatisierungsaufgaben stellt das SolutionCenter dar. Der C/C++ Developer (basierend auf dem Eclipse-CDT) ist zentraler Bestandteil dieses Engineering-Tools. Somit können alle Möglichkeiten des SolutionCenters wie z. B. Verlinkung von Projekten, Versionsverwaltung und der Eclipse-Marketplace genutzt werden. Die M1-Steuerungsfamilie stellt eine Multitasking-Umgebung auf Basis des Betriebssystems VxWorks® zur Verfügung. Jedes Software-Modul kann in beliebig viele Tasks gegliedert werden und dabei alle Möglichkeiten des Betriebssystems nutzen.



Projektvorlagen, sogenannte Templates, ermöglichen es, standardisierte Software-Module zu erstellen. Projekte werden als Basis für Templates genutzt. Diese Templates können anschließend mit einer für diese Aufgabe optimierten Sprache angepasst werden. Diese Sprache ermöglicht es, Ersetzungen durchzuführen, Dateien einzubinden oder Menüs zu erstellen, um das Template bei der Auswahl an eine bestimmte Aufgabe anzupassen. Die Templates können entweder als Datei eingebunden oder über den Eclipse Marketplace zur Verfügung gestellt werden.

Die eigentliche Programmierung wird durch einen modernen Editor erleichtert. Dieser Editor unterstützt IntelliSense (automatische Ergänzung von Strukturelementen, weitergehende Informationen zum Code-Fragment usw.), Syntax-Highlighting, faltbare Code-Bereiche usw.

Kompilieren und Linken kann automatisch durch das Tool oder gezielt manuell gesteuert werden (automatischer/manueller Build, Post-/Pre-Build). Das Ergebnis ist eine fertige, ausführbare Programmdatei für die M1-Steuerungsfamilie.

Viele Problemstellungen sind weitgehend unabhängig von der genutzten Plattform z. B. digitale Filter. Der C/C++ Developer stellt die Möglichkeit zur Verfügung auch für alternative Zielsysteme (z. B. MinGW, Windows usw.) lauffähige Software-Programme zu erstellen (alternative Launch-Konfigurationen). Dies ermöglicht ein effizientes Testen der Programme auch ohne Ziel-Hardware.

Für die Fehlersuche wird der Debugger über das grafische Interface bedient. Der verwendete Debugger (GDB) ermöglicht es, alle Bereiche des Programms im Schrittbetrieb zu untersuchen. Der Schrittbetrieb steht sowohl im C/C++ Bereich als auch im Assembler-Bereich zur Verfügung. Ansichten (Views) auf alle Bereiche des Programms (Speicher, Register, Variablen usw.) unterstützen bei der Fehlersuche. Abgerundet wird der Debugger durch die Möglichkeit mehrere Software-Module gleichzeitig zu debuggen und zu steuern.

Für den Automatisierungsbereich stellt der C/C++ Developer eine umfassende Lösung für die Programmierung und das Debuggen in C/C++ dar.

Editor

- Strukturierung von Projekten durch Unterverzeichnisse
- IntelliSense: automatische Ergänzung von Strukturelementen
- Syntax-Highlighting
- Faltbare Code-Bereiche, z. B. für komplette while-Schleifen
- Sprung zur Deklaration von Variablen und Funktionen aus dem Code

Build/Make

- Pre- und Post-Buildsteps
- Automatischer/manueller Make-Run

Compile

- gcc 2.95
- gcc 4.1.2

Debuggen

- Stepping (Step-Into, Step-Over, Step-Return)
- Views (Ausdrücke, Speicher, Disassembly)
- Assembler-Stepping
- Debuggen von mehreren Software-Modulen gleichzeitig
- Aufrufbaum

Allgemein

- Anbindung an den Marketplace
- Basiert auf der aktuellen CDT-Version
- Toolchain beherrscht auch MinGW, Cygwin und Windows usw. (Erstellen/Testen von Programmen ohne Zielhardware)
- Verwalten unterschiedlicher Konfigurationen
- Installation und Start des erstellten Software-Moduls aus der Entwicklungsumgebung
- Ankopplung des Debuggers auf laufende Software-Module

Features

- Erstellen von Externen Bibliotheken für PLC-Projekte
- Versionsverwaltung über SVN oder GIT
- Verlinkung von Projekten
- Einbindung von 3rd Party Plugins bzw. externen Plugins
- *Templates:*
 - Geführtes Erstellen von Templates
 - Anpassung des Templates an die eigenen Bedürfnisse durch eine optimierte Template-Sprache