

Historie

EasyCODE war eines der ersten Entwicklungstools, das für prozedurale Programmiersprachen eine grafische Entwicklungsoberfläche anbot und dafür konsequent auf die Vorteile von Struktogrammen nach Nassi-Shneiderman setzte, die sich in ungezählten Projekten bewährt, und bis heute nichts an Aktualität und Effizienz verloren haben. Bereits in den 90er Jahren bot SIEMENS eine erste Version an, mit der es möglich war aus Struktogrammen SCL-Dateien zu generieren. Manche unserer Kunden schreiben Ihre SPS Programme noch heute damit.

Motivation

EASYCODE® ist in der klassischen Embedded Entwicklung zu einem Standard geworden. Natürlich entwickeln EasyCODE Kunden auch Lösungen und Produkte die mit SPSen arbeiten und wir sehen diese Bereiche immer mehr zusammen wachsen. Embedded- und SPS-Entwicklungsteams arbeiten oft gemeinsam an einem Projekt. So kommen SPS-Entwickler mit EasyCODE in Berührung und lernen die Vorzüge strukturierter Entwicklung kennen und schätzen.

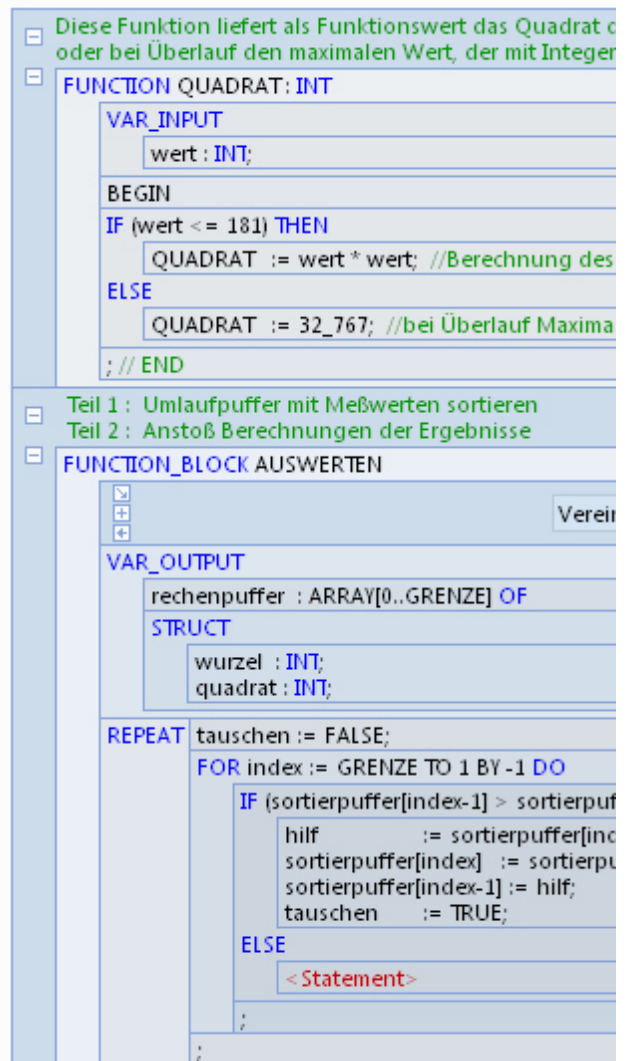
Die textbasierten Programmiersprachen der IEC 61131-3 eignen sich hervorragend für strukturierte Programmierung. Mit diesen Voraussetzungen war es nahe liegend, eine EasyCODE Version für Entwicklung und Analyse von SPS Programmen anzubieten.

Philosophie & Technik

Bei Struktogrammen spricht man oft auch von Code-Generatoren, was bei EasyCODE SPS jedoch nicht zutrifft. EasyCODE SPS liest Quellcode ein, interpretiert diesen und schreibt beim Speichern wieder reinen Quellcode. Struktogramme sind nur eine andere Art der Darstellung von Sprachkonstrukten die wir in allen Hochsprachen finden. Diese grafische Darstellung bietet einige Vorteile bei der Erstellung neuer Dateien. Man beginnt die Entwicklung einer Datei in grafischen Symbolen, z.B. mit einem Anweisungsblock, hängt einen Funktionsblock an und beschreibt verbal, was diese Funktion machen soll, arbeitet die ersten Details aus und hängt den nächsten Funktionsblock an. So entsteht schnell ein Gerüst das schon die Logik enthält. Dieses Konstrukt wird verfeinert, indem man die Struktogrammelemente mit Quellcode füllt.

Ein großer Vorteil dieser Art der Entwicklung ist, dass man nicht an starre Vorgehensweisen gebunden ist, sondern sehr flexibel auf geänderte Anforderungen reagieren kann. Designfehler in einer Datei sind schnell korrigierbar, weil das betreffende Element einfach per Drag & Drop an die gewünschte Position verschoben werden kann. Funktionen die in anderen Steuerungen genauso oder mit kleinen Modifikationen benötigt werden, können in EasyCODE als Bausteine abgelegt und fertig strukturiert am Verwendungsort eingefügt werden.

Eine der großen Stärken von EasyCODE ist die Option, Dateien weiter zu strukturieren um die Lesbarkeit zu erhöhen und die Konzentration auf das Wesentliche zu lenken. Wir sprechen von Ebenen oder Ausblendungen, wenn Teile des Quellcodes aus der aktuellen Ansicht ausgeblendet und in eine neue Ebene verschoben werden. Mit dieser



Technik kann die Logik einer Funktion sehr gut herausgearbeitet werden und man erreicht zwangsläufig, dass auf jeder Seite eine gut lesbare, verständliche Einheit abgebildet wird.

Analyse komplexer Dateien

Easy**CODE** eignet sich hervorragend zur Analyse komplexer Dateien und Algorithmen, da es Quelldateien beim Einlesen interpretiert und in Struktogramme wandelt. Funktionsblöcke, Daten- und Organisationsbausteine werden beim Öffnen der Datei ausgeblendet. Ähnlich dem Inhaltsverzeichnis eines Buches erhält man so in der Strukturliste eine Gliederung und kann daraus gezielt einzelne „Kapitel“ ansteuern. Wechseln Sie in die Detailansicht um den Quellcode zu inspizieren, oder innerhalb des Elements logische Einheiten zu bilden und die Struktur zu optimieren. Wenn reiner Quelltext eingefügt wird, baut Easy**CODE** daraus ein Struktogramm, das sich nahtlos in die Datei integriert.

Zur Weiterentwicklung bietet Easy**CODE** sprachspezifisch alle möglichen Elemente aus dem Kontextmenü an und verhindert dabei Einfügeoperationen entgegen der Sprachdefinition. Routineaufgaben wie Klammern und Einrückungen zählen, Zeilenabschlüsse setzen und Kommentierungen erledigt Easy**CODE** automatisch und syntaktisch richtig. Eine Metrikanzeige warnt den Entwickler wenn die Komplexität der Datei einen kritischen Wert erreicht, die Entscheidungsdichte das Programm schlecht wartbar werden lässt oder einfach nur ein paar Kommentarzeilen gefordert sind.

Dokumentation

Mit Struktogrammen haben Sie immer eine zum Quellcode konsistente, ISO 9001 konforme Dokumentation. Nicht nur der TÜV fordert Struktogramme oft als wesentlichen Teil einer Dokumentation. Easy**CODE** bietet die Möglichkeit, Struktogramme wie dargestellt an einen Drucker zu senden oder einzelne Elemente daraus in andere Applikationen wie MS Word zu übernehmen um individuellen Anforderungen an die Dokumentation gerecht zu werden.

Projektunterstützung

Easy**CODE** bietet eine sehr komfortable Projektumgebung zur einfachen Übernahme von Bestandsprojekten, die dem Windows-Explorer nicht unähnlich ist. Es erleichtert die Re-Strukturierung von Projekten erheblich, bietet eine Schnittstelle zur Integration externer Compiler und stellt eine programmierbare Anbindung an Versionskontrollsysteme bereit.

Wann und warum sollte man EasyCODE einsetzen?

SIEMENS® empfiehlt für die Entwicklung komplexer Steuerungen SCL (Structured Control Language) als die Hochsprache, mit der selbst komplizierte Aufgaben problemlos zu bewältigen sind. Zitat: „S7-SCL® eignet sich insbesondere für die Programmierung von komplexen Algorithmen und mathematischen Funktionen oder für Aufgabenstellungen aus dem Bereich der Datenverarbeitung.“

Dem schließen wir uns vorbehaltlos an und empfehlen Easy**CODE** immer dann zu verwenden, wenn Sie in Hochsprachen programmieren können. EASYCODE® versteht sehr viel davon und gewährleistet, dass Sie es auch einsetzen können, wenn Sie Ihre SPS eines Tages in C oder C++ programmieren. Aber auch ellenlange Anweisungslisten können mit Easy**CODE** logisch strukturiert und damit deutlich lesbarer gemacht werden.

Wann rechnet sich EasyCODE?

Die Erfahrung zeigt, dass in der Implementierungsphase eine Zeiteinsparung von 40% gegenüber einer Entwicklung in textuellen Systemen keine Seltenheit ist. Durch den strukturierten Aufbau der Dateien sinkt die Fehlerwahrscheinlichkeit im Design, der Editor verhindert syntaktische Fehler, bietet innovative Techniken zum Rapid-Coding, unterstützt bei der Analyse komplizierter Dateien und liefert eine vollständige, normgerechte Dokumentation.