



Datenblatt CODESYS Application Composer

Der CODESYS Application Composer ist ein Entwicklungstool zur effizienten Erstellung von Applikationen die aus wiederkehrenden Funktionsbausteinen bestehen.

Zum Test ist eine 30 Tage Demo Lizenz verfügbar. Für die Vollversion kontaktieren Sie bitte den CODESYS Vertrieb.

Produktbeschreibung

Mit dem CODESYS Application Composer können komplette Steuerungsapplikationen aus zuvor erstellten Softwaremodulen konfektioniert, das heißt zusammengestellt und parametrierung werden.

Module sind funktionale Programmeinheiten, die sowohl Maschinen- oder Anlagenteilen als auch Softwarefunktionen entsprechen können. Sie enthalten neben dem Programmcode typischerweise Komponenten wie Visualisierungselemente, Parametrierung oder E/A-Zuordnungen. Sie decken damit nahezu alle Programmier-Aspekte ab, die CODESYS bietet.

Die Konfektionierung der Module erfolgt im Modulbaum. Jeder Eintrag entspricht dabei einer Modulinstanz. Beim Einfügen neuer Elemente in die Modulstruktur werden nur passende Module zur Auswahl angeboten. In den Moduleigenschaften werden die Parametrierung, die E/A-Konfiguration und die Visualisierungsauswahl für die Modulinstanzen festgelegt. Gleichzeitig kann die Konfiguration von Schrittkettenmodulen mit Hilfe eines komfortablen Ablaufeditors erfolgen. Aus der Modulkonfiguration wird über ein Menükommando der komplette Applikationscode inklusive Visualisierung und E/A-Konfiguration erzeugt. Applikationsspezifischer Code kann in Form von Erweiterungsmodulen hinzugefügt werden, ohne bei einem erneuten Generatorlauf verlorenzugehen

Mit dem Kauf einer funktionsfähigen Lizenz erwerben Sie die Möglichkeit innerhalb des CODESYS Development Systems Module zu erstellen und zu verwenden.

Das Anlegen von Modulen erfolgt in Form von Moduldeklarationen, die im POU-Pool als Objekte hinzugefügt werden können (siehe Bild 1).

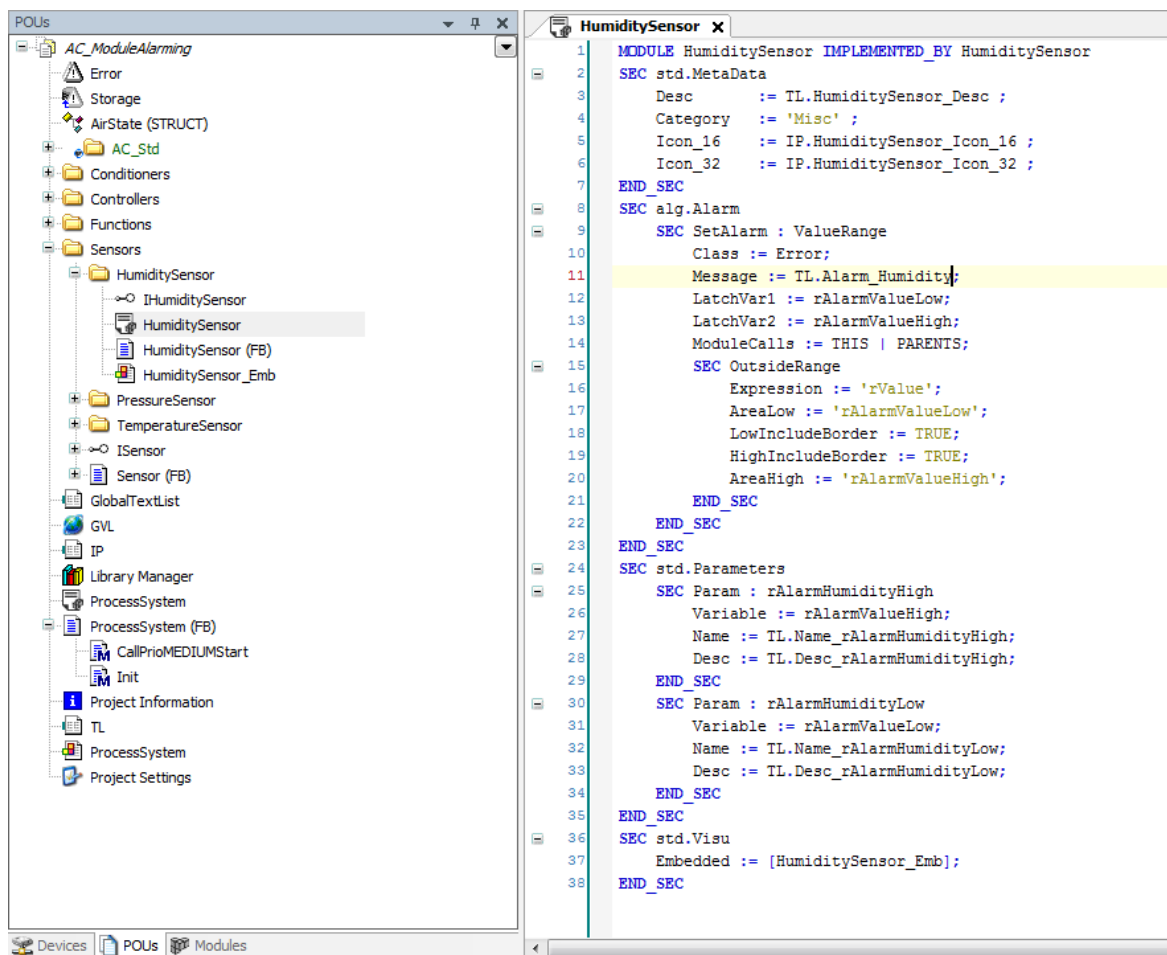


Bild 1: Moduldeklaration

Jede Moduldeklaration verlangt als Basis die Angabe eines Funktions-Bausteins (Modul-FBs), der die programmatischen Funktionalitäten des Moduls implementiert. Innerhalb der Moduldeklaration kann der Modul-FB durch zusätzliche Eigenschaften und zugehörige Objekte ergänzt werden:

- **Parameter:** Input-Variablen des Modul-FBs können als Parameter gekennzeichnet werden. Parameter werden dann komfortabel innerhalb eines Parameter-Modul-Editors konfiguriert (siehe Bild 2).

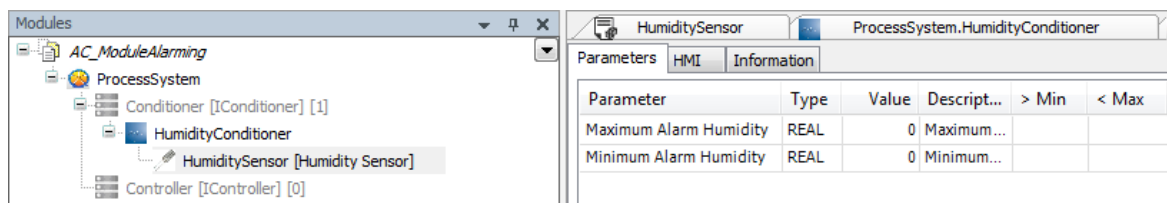


Bild 2: Parameter-Editor

- **Eingänge/Ausgänge:** Input- und Output-Variablen des Modul-FBs können als Modul-Ein- und Ausgänge definiert werden. Die Modul-Ein-/Ausgänge können dann komfortabel mit Variablen, anderen Modul-Ein-/Ausgängen oder Geräte-Ein-/Ausgängen verbunden werden (siehe Bild 3).

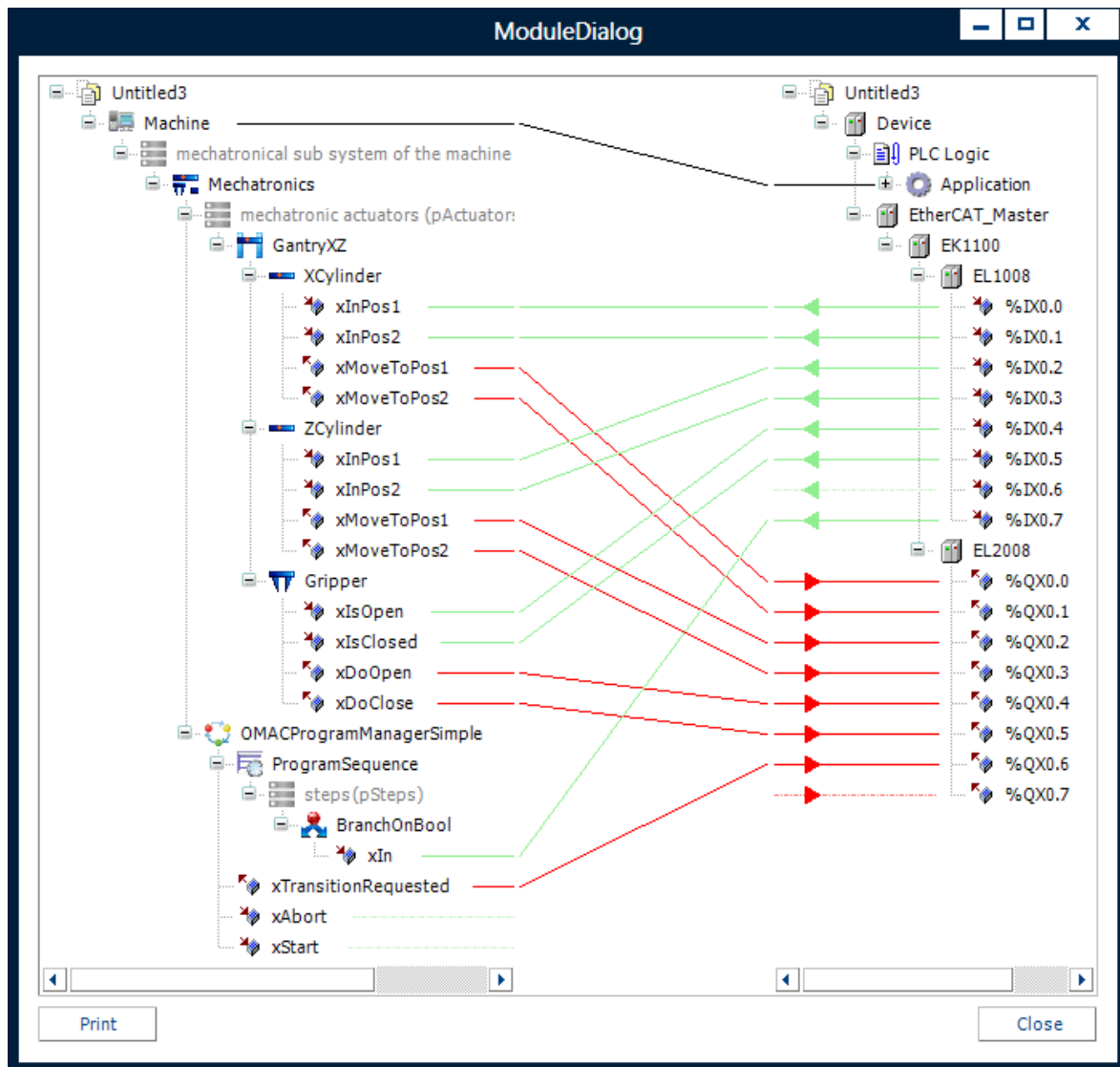


Bild 3: E/A-Editor

- Slots: Input-Variablen des Modul-FBs können so gekennzeichnet werden, dass sie Instanzen anderer Modul-FBs aufnehmen können. Soll beispielsweise ein Modul unterhalb eines anderen Moduls als Sub-Modul (Kind) einsetzbar sein, so wird ein Slot im Vater-Modul definiert, der die entsprechenden Kind-Module aufnehmen kann. Die zugehörige Input-Variable wird dann bei der Generierung automatisch mit den Instanzen der Kind-Modul-FBs befüllt.
- Tasks: Handelt es sich um ein Toplevel-Modul (Modul ohne Vater-Modul), so können zusätzlich Tasks definiert werden, die beim Generatorlauf erzeugt werden und bestimmte Methoden des Modul-FBs automatisch aufrufen. Ein Toplevel-Modul ruft dann seinerseits Methoden seiner Kind-Modul-FBs auf.
- Visualisierungen: Jedes Modul erlaubt es Seiten- und eingebettete Visualisierungen zu definieren. Diese werden beim Generieren automatisch mit dem Modul erzeugt und verbunden. Seitenvisualisierungen werden explizit für ein Modul angezeigt. Eingebettete Visualisierungen können dagegen in die Seitenvisualisierungen von Vater-Modulen eingebettet werden.
- Proxy-Modul-FBs: Module können sogenannte Proxy-Vertreter-FBs ihres eigenen Modul-FBs definieren. Ein Proxy dient dazu Referenzen auf einen Modul-FB, die über

Applikations- und Steuerungsgrenzen hinaus gehen, zu ermöglichen, indem als Vertreter des referenzierten Modul-FBs ein Proxy-FB in der jeweiligen Ziel-Applikation angelegt wird. Die Kommunikation und der Datenaustausch zwischen Modul-FB und seinem Proxy-FB unterhalb einer fremden Applikation wird automatisch von den Application Composer Generatoren erzeugt.

- **Instanz-Referenzen:** Mit sogenannten Instanz-Referenzen ist es möglich FB-Instanzen zu definieren, die erst zur Konfigurations-Zeit der Module vom Modul-Benutzer mit tatsächlichen Instanzen identifiziert werden. So können z.B. Geräte-FBs in Modulen als Instanz-Referenzen referenziert werden.
- **Geräte und Ein-/Ausgänge:** Module können Geräte (z.B. Feldbusse), definieren, die mit dem Modul in den Gerätebaum eingefügt werden. Die Ein- und Ausgänge der Geräte können dann automatisch mit den Ein- und Ausgängen des Moduls, bzw. der Module, verbunden werden. Für eine möglichst große Flexibilität können die Geräte auch als so genannte „Wildcards“ eingefügt werden, die erst zum Generierungs-Zeitpunkt mit tatsächlichen Gerätetypen befüllt werden müssen (siehe Bild 4).

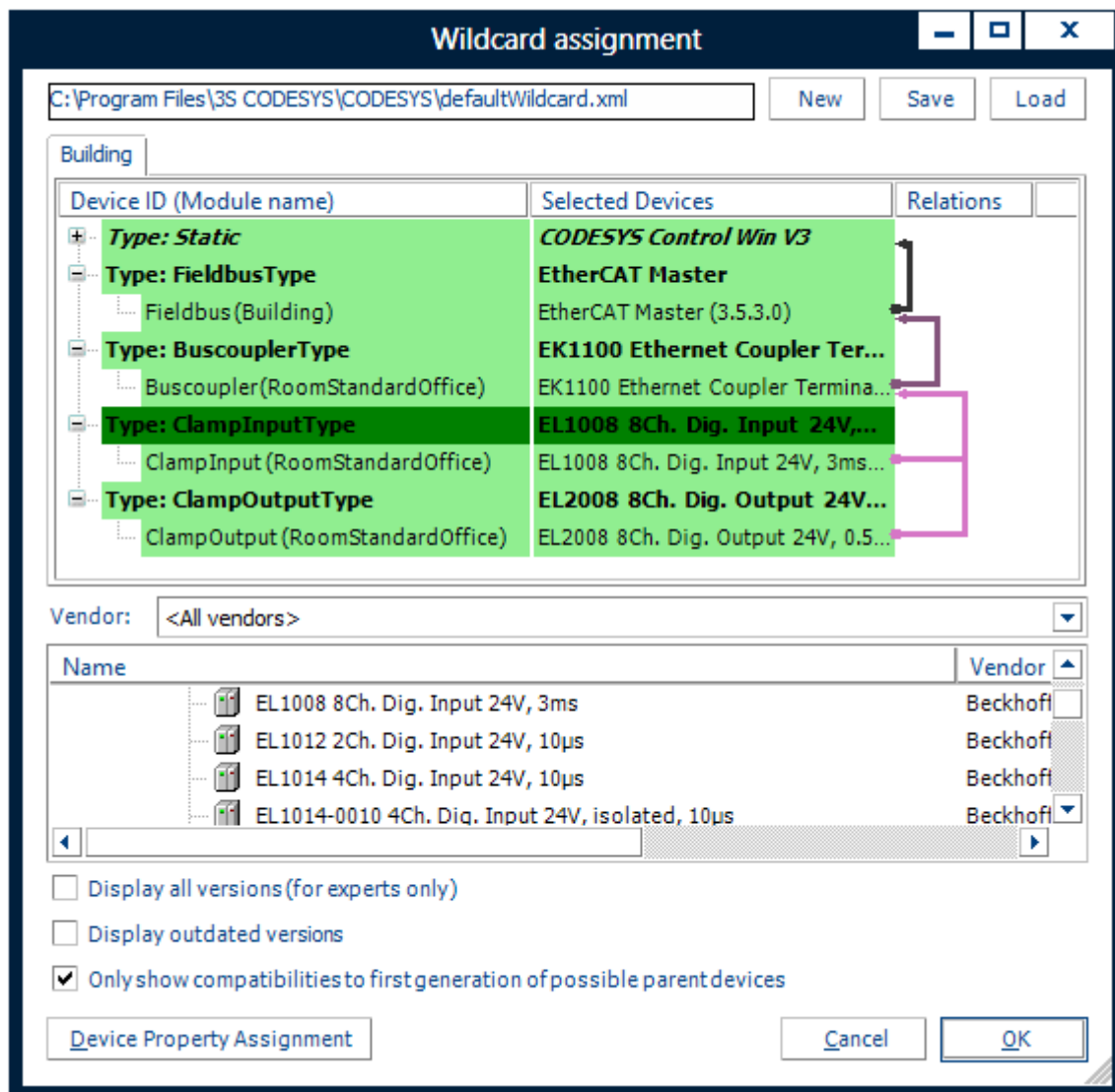


Bild 4: Zuordnung der Modulkanäle zu Geräten

- **Alarmer:** Neben Visualisierungen und Geräten, können Module auch das CODESYS-Alarm-Management verwenden, um Alarmer für Variable ihrer Modul-FBs zu erzeugen. Diese

Alarmer können dann sowohl angezeigt, als auch über einen Aufrufmechanismus abgefangen und ausgewertet werden.

- Ablauf-Modul-Kennzeichnung: Wenn Module in Form von Sequenzen editierbar sein sollen, kann dies über die Modul-Deklaration definiert werden. Es können Parameter, Ein-/Ausgänge oder Referenzen angegeben werden, die dann direkt in den Ablaufschritten angezeigt werden.
- Default-Submodule: Für Modul-Slots (siehe oben), können Default-Belegungen und Default-Konfigurationen angegeben werden. Eine solche Default-Belegung füllt dann beim Einfügen des Moduls den Modul-Slot automatisch mit dem vordefinierten Modul, welches entsprechend konfiguriert werden kann. Nach der Deklaration der Module und dem Implementieren des zugehörigen Modul-FBs, können diese Module im Modulbaum eingefügt und konfiguriert werden. Um nun schließlich eine funktionsfähige IEC-Applikation zu erhalten muss nur noch ein Generator-Lauf durchgeführt werden. Bei diesem werden, je nach Auswahl der Generatoren und Konfiguration, der komplette IEC-Code, Visualisierungen, Geräte usw. unterhalb einer Applikation erzeugt. Der gesamte erzeugte Code, bzw. alle dabei angelegten Objekte, sind unter dem Gerätebaum frei einsehbar und editierbar.

Allgemeine Informationen

Lieferant:

CODESYS GmbH
 Memminger Straße 151
 87439 Kempten
 Deutschland

Support:

Technischer Support ist bei diesem Produkt nicht enthalten. Um technischen Support zu erhalten, erwerben Sie bitte ein CODESYS Support Ticket.

<https://support.codesys.com>

Artikelname:

CODESYS Application Composer

Artikelnummer:

2101000006

Vertrieb/Bezugsquelle:

CODESYS Store
<https://store.codesys.com>

Lieferumfang:

- Lizenzschlüssel

Systemvoraussetzungen und Einschränkungen

Programmiersystem	CODESYS Development System 3.5.17.30 oder höher
Laufzeitsystem	CODESYS Control Version 3.5.0.0
Unterstützte Plattformen/ Geräte	Hinweis: Verwenden Sie das Projekt <i>Device Reader</i> , um die von der Steuerung unterstützten Funktionen zu ermitteln. <i>Device Reader</i> ist kostenlos im CODESYS Store erhältlich.
Zusätzliche Anforderungen	-
Einschränkungen	-
Lizenzierung	• Soft Key (Arbeitsplatzgebundene Lizenzierung, kostenloser Bestandteil aller CODESYS Produkte) • Optional: CODESYS Key

(Erhöhte Sicherheit gegen Verlust der
Lizenzschlüssel, Übertragbare Lizenzierung auf
andere Arbeitsplätze)

Erforderliches Zubehör

Optional: CODESYS Key

Bitte beachten Sie: Technische Änderungen, Druckfehler und Irrtümer vorbehalten. Es gilt der Inhalt der aktuellen Online-Version dieses Dokuments.

Erstellungsdatum: 13.07.2023