



MATCH Project Definition Tool

Functional Safety
SIL 2
PL d
AgPL d (SRL 2)

Besondere Merkmale

- **SIL 2 / PL d / AgPL d zertifizierte Software**
- Software-Projektdefinition auf Fahrzeug- bzw. Maschinen-Ebene
- Projekt-Unterstützung für mehrere Steuergeräte
- Automatische Code Generierung für Steuergeräte und Displays
- Sicherheitsgerichteter Entwicklungszyklus nach V-Modell wird unterstützt
- Erfassung von Requirements sowie Use-Case- und Test-Case-Verwaltung
- Generierung der Projekt- und Entwicklungs-Dokumentation unter Einbindung eigener Templates
- Umfassende Definition der:
 - CAN-Kommunikation
 - Fehler-Meldungen und -Reaktionen
 - Parameter- und Options-Listen in NvMem- und Flash-Speicher
 - E/A-Pins unter Berücksichtigung der Sicherheitsanforderungen
- Verwendung zertifizierter Treiberbausteine zur Ansteuerung angeschlossener Aktoren und Sensoren
- Generierung von embedded Code für unterstützte Steuergeräte-Hardware, sowie Code zur PC-Simulation der Applikation (Software in the Loop)
- Unterstützung von Modul- und Integrations-Tests
- Toolbox-Schnittstelle zum flexiblen Einbinden zertifizierter Bibliotheksbausteine

Beschreibung

Das **Project Definition Tool (PDT)** ist eine PC Software zur Entwicklung von Applikationen (embedded Software) für komplexe Fahrzeug- bzw. Maschinen-Steuerungen bestehend aus einem oder mehreren Steuergeräten und/oder Displays.

Das PDT besteht im Wesentlichen aus:

- Graphischer Oberfläche zur Eingabe und Pflege der Daten,
- Auto-Code-Builder für Steuergeräte- und Display-Software sowie
- HYDAC MATCH Core Software und Standard-Toolbox

Es sind drei PDT-Grundversionen verfügbar:

- **Programmer C:** Paket zur Programmierung einzelner Steuergeräte in "C", für allein arbeitende Entwickler
- **Programmer CODESYS*:** Paket zur Programmierung einzelner Steuergeräte und Displays in CODESYS, für allein arbeitende Entwickler
- **Standard Developer:** Komplettpaket zur System-Software-Entwicklung vollständiger Maschinen durch Entwickler Teams, System-Spezialisten und Requirement Manager

Nutzen Sie die PDT-Oberfläche im sicherheitsgerichteten Entwicklungszyklus auf Fahrzeugebene für folgende Arbeitsschritte:

- Definieren der Anforderungen und Softwarespezifikationen
- Anlegen des System-Designs mit Steuergeräten und Displays
- Konfigurieren der Pins, CAN-Nachrichten sowie des fahrzeugübergreifenden Fehlermanagements
- Einstellen der Bibliotheks-Bausteine (Blöcke) zur Ansteuerung der angeschlossenen Sensoren und Aktoren
- Einrichten der Datenbasis für Parameter, Teach-Werte und Optionen

Der Einsatz getesteter und zertifizierter Toolboxes vereinfacht und beschleunigt dabei die Entwicklung der Applikationen wesentlich. Nach erfolgter Systemdefinition generiert das PDT aus dem Projekt – je nach gewählter Programmiersprache – mit dem Auto-Code-Builder zertifizierten Code (nach SIL 2, PL d und AgPL d), den so genannten MATCH-Software-Rahmen. Dieser Software-Rahmen bildet die Grundlage für die Applikationsentwicklung, wobei sich der Entwickler auf die Funktionen der entsprechenden Maschine beschränken kann und von Standard-Software-Funktionalitäten weitgehend entlastet wird.

Zusätzlich erzeugt das PDT eine vollständige Projektdokumentation aus den Eingaben und stellt die Schnittstellen zum Wartungswerkzeug MST (**M**aschinen **S**ervice **T**ool) und zum Testwerkzeug TSE (**T**est and **S**imulation **E**nvironment) bereit.

* in Vorbereitung

Technische Daten

Systemvoraussetzungen Software

Unterstützte Betriebssysteme Windows® 7, 8 oder 10 (32-/64-bit)

Weitere Software .NET 4.6 Framework,
C-Compiler für die entsprechenden Steuergeräte
PDF-Anzeigeprogramm wie z. B. Adobe Acrobat® Reader®

Systemvoraussetzungen Hardware

Prozessor minimal 2 Kernprozessor mit 1,6 GHz

Speicherbedarf RAM minimal 2 GB (empfohlen 4 GB oder mehr)

Speicherbedarf Festplatte minimal 1 GB freier Speicher

Bildschirmauflösung minimal 1.024 x 768

Erfüllt folgende Standards

Funktionale Sicherheit IEC 61508 – SIL 2, EN ISO 13849 – PL d,
ISO 25119 & EN 16590 – AgPL d (SRL 2)

Auto Code Generierung

Steuergeräte Programmiersprache C

Display Programmiersprache CODESYS 3.5

Steuergeräte	Non safety	Safety
	HY-TTC 30-H, HY-TTC 30-I, HY-TTC 32-H, HY-TTC 50, HY-TTC 60	HY-TTC 30S-H, HY-TTC 32S-H, HY-TTC 90, HY-TTC 94, HY-TTC 71, HY-TTC 77, HY-TTC 510, HY-TTC 540, HY-TTC 580
Displays	Touchscreen	Ohne Touchscreen
	HY-eVision² 7.0 und HY-eVision² 10.4	HY-eVision² 7.0

Lieferumfang*

Installationsprogramm für:

- PDT Desktop Anwendung,
- MATCH Core für die Steuergeräte und Displays sowie die PC-Simulation,
- Standard-Bibliotheken,
- Standard-Toolbox sowie
- Integrierte Entwicklungsumgebung (Eclipse)

Zubehör

(nicht im Lieferumfang enthalten, bitte extra bestellen)

- Kabelbäume für Steuergeräte
- Programmierkabel (CAN)
Materialnummer: 6149786
für HY-TTC 50/60/90/94 oder
- Programmierkabel (CAN)
Materialnummer: 61499787
für HY-TTC 77
- PCAN-Dongle ZBS PCAN-USB
Steckverbinder
Materialnummer: 6163719
- Wartungs- und Test-Werkzeuge
(MST, TSE)

Typenschlüssel Grundversionen

MATCH PDT – XX – G10 – 000

Programm-Variante

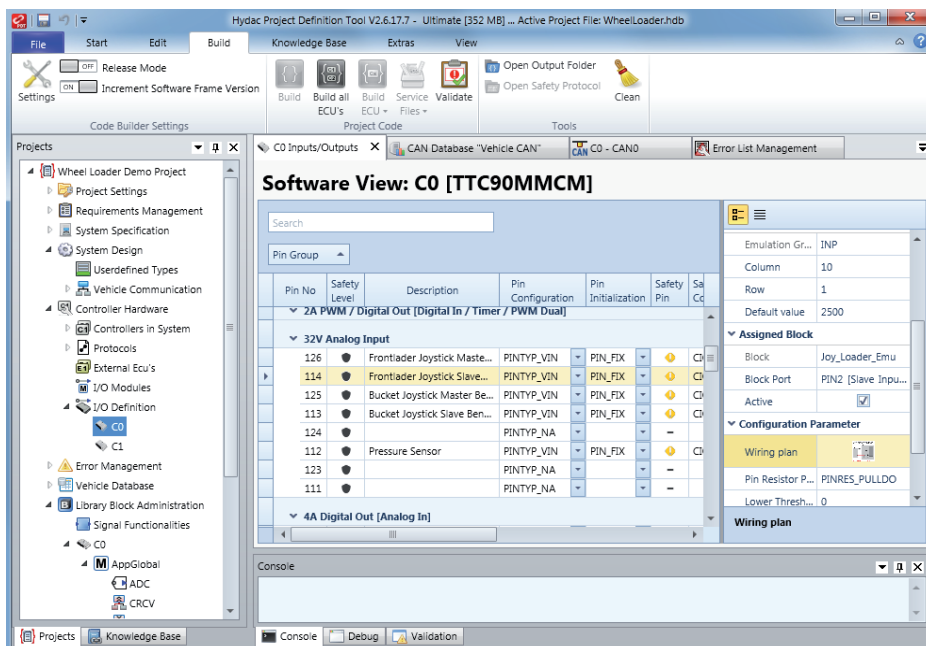
00 = Standard Developer
CP = Programmer C
CD = Programmer CODESYS

Software-Version

G10 = aktuelle Version

Modifikationsnummer

000 = Standard



Screenshot des PDT – Pin Konfiguration

* je nach geordneter Option und Output-Format

PDT Add-Ons

Ein Add-On ist eine Erweiterung des PDT-Funktionsumfangs.

Typenschlüssel PDT-Add-On

MATCH PDT AddOn – **XX** – **G10** – **YYYY** – **000**

Programm-Variante

00 = Standard Developer
 CP = Programmer C
 CD = Programmer CODESYS

Software-Version

G10 = aktuelle Version

Erweiterungs-Variante

CEN = Automatic Code Generation of embedded C-Code
 CCG = C Code Generation
 S2Pd = Safety Certified Code Generation (SIL 2, PL d, AgPL d)
 DCD3 = V²-MATCH Library – CODESYS 3.x Code Generation
 CANO = CANopen Stack
 SPCK = Specification Package
 RCAP = Requirements Capturing
 ISOB = ISOBUS Plug-in
 UDSB = UDS Basic – Basic UDS Communication Interface
 ISYM = SYM-File Import/Export Interface
 RQIF = Requirements ReqIF-Schnittstelle
 DBCE = DBC export
 DMSY = Document Management System
 HMG4 = HMG 4000 Integration

Modifikationsnummer

000 = Standard

Die für die unterschiedlichen Programm-Versionen verfügbaren Add-Ons zeigt folgende Tabelle. Für verfügbare Optionen wurde die zugehörige Materialnummer eingetragen. Nicht verfügbare Optionen sind mit „-“ und bereits enthaltene Add-Ons sind mit „✓“ gekennzeichnet.

Leistungsmerkmale	Standard Developer	Programmer C	Programmer CODESYS
Mehrere Steuergeräte	✓	9576 (MECU)	9645 (MECU)
Sicherheitszertifizierter Code (SIL 2, PL d, AgPL d)	✓	9509 (CEN)	9641 (S2Pd)
C Code Generation	9588 (CCG)	✓	-
Display Bibliothek V ² -MATCH – CODESYS 3.5 Display Code	9501 (DCD3)	9577 (DCD3)	✓
Allgemeine Projektdokumentation als PDF	✓	✓	✓
Dokumenten-Management	✓	9579 (DMSY)	9646 (DMSY)
Erweiterte Spezifikationen im kundenspezifischen Design	9503 (SPCK)	-	-
Requirements Management	9505 (RQIF)	-	-
Requirements ReqIF-Schnittstelle	9507 (RCAP)	-	-
Diagnose-Schnittstelle „UDS Basic“	✓	9511 (UDSB)	9642 (UDSB)
CANopen Stack	9502 (CANO)	9510 (CANO)	✓
Symbol-Datei Import-/Export-Schnittstelle	✓	9512 (ISYM)	9647 (ISYM)
ISOBUS Plug-in	9508 (ISOB)	-	-
Start Seiten Editor	✓	-	-
Mehrsprachige Informations-Ausgabe	✓	-	-
DBC Export	9569 (DBCE)	9580 (DBCE)	9648 (DBCE)
HMG 4000 Integration	✓	9513 (HMG4)	9643 (HMG4)

Toolboxen

Eine Toolbox ist eine Sammlung von Bibliotheksbausteinen. Sie besteht aus Blöcken und Signalgliedern. Alle Bausteine sind nach den oben genannten Standards der Funktionalen Sicherheit zertifiziert.

Signalglieder

Signalglieder sind Software-Bausteine, die meist Grundfunktionen für Ihre Applikation bereitstellen. Sie sind vollständig gekapselt und können eigene Zustandsgrößen enthalten. Sie verwenden Signalglieder im „C“-Code ähnlich wie komplexe Variablen. Sie initialisieren die Signalglieder mit Zugriffsfunktionen und können diese damit auch in der Applikation anpassen bzw. benutzen.

Blöcke

Blöcke werden im PDT angelegt und über den Auto-Code-Builder mit dem MATCH-Software-Rahmen generiert. Blöcke können innerhalb des PDTs mit Eingangs-/Ausgangs-PINs der Steuergeräte verbunden werden.

Es gibt Blöcke, die ein oder mehrere Eingangs-Pins auslesen (Treiber für z. B. Sensoren, Schalter ...), Blöcke, die einen oder mehrere Ausgangs-Pins steuern (Treiber für spezifische Aktoren) und Blöcke zur z. B. Regelung ohne Hardware-Zugriff.

Eingangsblöcke ermöglichen z. B. eine Frequenz-, Strom- oder Spannungsmessung mit entsprechender Fehlererkennung bzw. Diagnosemöglichkeiten.

Ausgangblöcke steuern z. B. unterschiedliche Ventiltypen, Lampen und andere elektrische Aktoren, wobei auch hier die entsprechenden Diagnosefunktionalitäten integriert sind.

Jeder Block kann:

- eine Fehlererkennung beinhalten und so mit dem Fehlermanagement verbunden sein
- mit einem oder mehreren Eingangs-/Ausgangs-Pins oder keinem Pin verbunden sein
- mit Datenbanklisten verbundene Parameter verwenden
- auf Signalgliedern aufbauen.

Typenschlüssel MATCH Toolboxen

MATCH ToolBox – XXX – G10 – YYY – 000

Steuergeräte-Familie

ALL = alle unterstützten Steuergeräte-Plattformen
 030 = HY-TTC 30-H, HY-TTC 30S-H
 032 = HY-TTC 32-H, HY-TTC 32S-H
 050 = HY-TTC 50, HY-TTC 60
 071 = HY-TTC 71
 077 = HY-TTC 77
 090 = HY-TTC 90, HY-TTC 94
 500 = HY-TTC 510, HY-TTC 540, HY-TTC 580

Software-Version

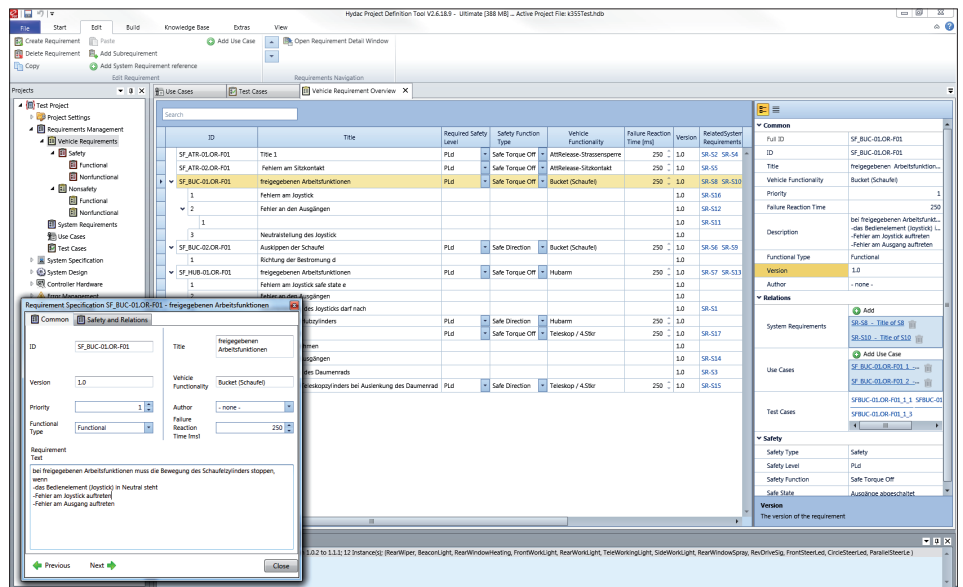
G10 = aktuelle Version

Toolbox-Variante

SCT = Signal Control Toolbox, weitere Toolboxen auf Anfrage

Modifikationsnummer

000 = Standard



Definition der Maschinen-Requirements mit dem PDT

Anmerkung

Die Angaben in diesem Prospekt beziehen sich auf die beschriebenen Betriebsbedingungen und Einsatzfälle.

Bei abweichenden Einsatzfällen und/oder Betriebsbedingungen wenden Sie sich bitte an die entsprechende Fachabteilung.

Technische Änderungen und Korrekturen sind vorbehalten.

HYDAC ELECTRONIC GmbH

Hauptstraße 27, 66128 Saarbrücken
 Tel. +49 6897 509-01
 Fax +49 6897 509-1726
 E-Mail: electronic@hydac.com
 Internet: www.hydac.com