

---

# EasyLab: Modell-basierte Software-Entwicklung für mechatronische Systeme

---

Prof. Dr.-Ing. habil. Alois Knoll (k@tum.de)  
Lehrstuhl für Echtzeitsysteme und Robotik  
Institut für Informatik  
Technische Universität München

# Überblick

- BMBF-Projekt EasyKit
- EasyLab
  - Anforderungs-Analyse
  - Modellierung
  - Modularer Aufbau
  - Parametrisierung und Debugging
- Ausblick

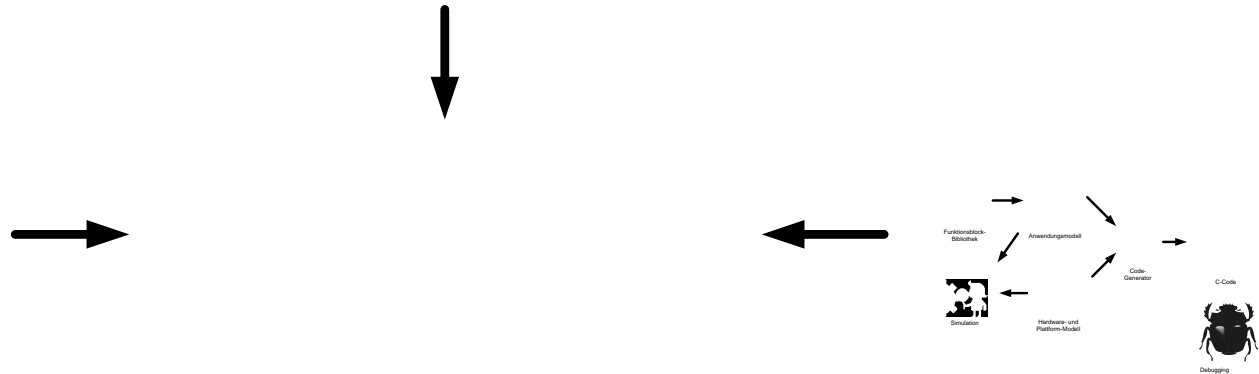
# BMBF-Projekt EasyKit

- Forschungskonzept „Forschung für die Produktion von morgen“
- Schwerpunkt: „Einführungsstrategien mechatronischer Systeme“
- Themenstellung „Zuverlässige mechatronische Systeme“

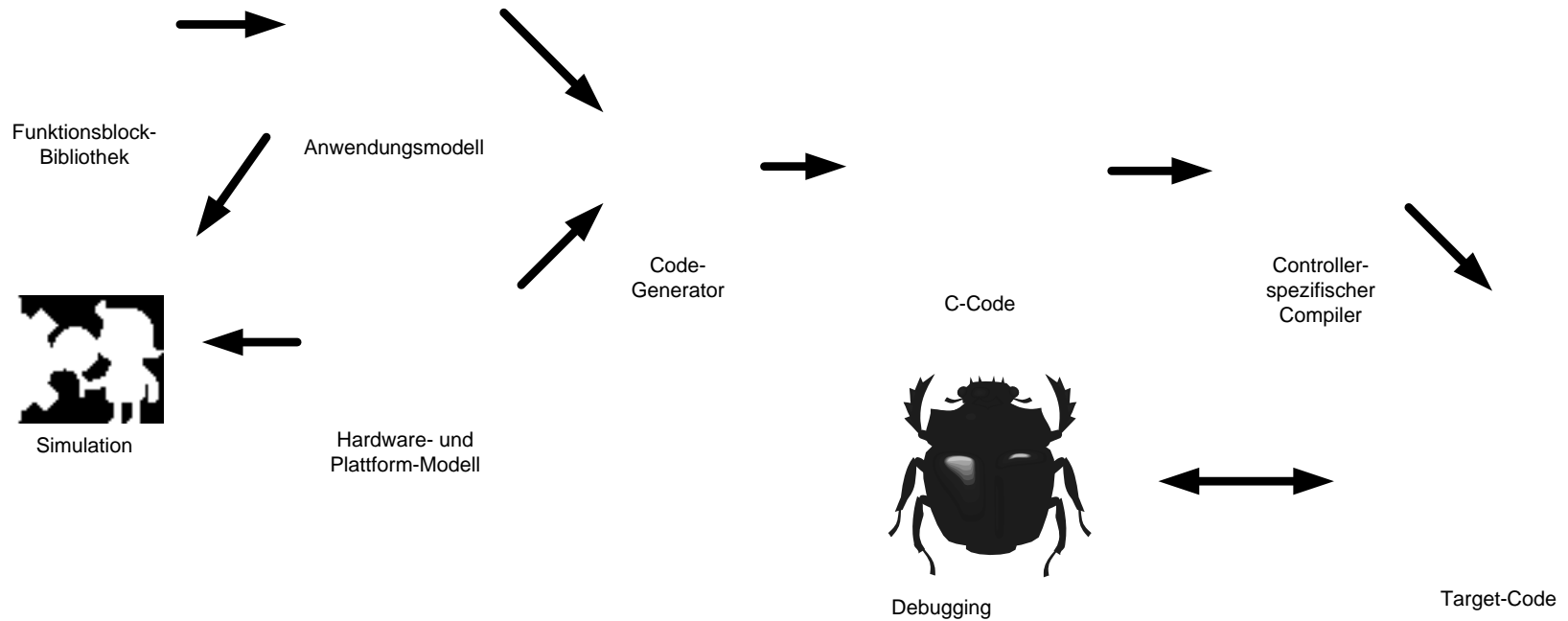
# Projektziele

- Ausgangspunkt: Match-X
  - ausgefeiltes Komponentensystem für Mechatronik
  - Problem
    - Benötigte Hardware kann auf einfache Weise in sehr kurzer Zeit bereitgestellt werden
    - Erstellung der benötigten Software ist bislang langwierig und erfordert Spezialisten
- Zielstellung:  
*„Modulare Software für modulare Hardware“*

# Von der Idee zum Mechatronischen System



# EasyLab - Überblick



# EasyLab - Überblick

- Anforderungen
  - Möglichst einfache, jedoch ausdrucksstarke Modellierung
  - Erstellung von Software für mechatronische Systeme auf Basis der Match-X-Bausteine (nicht ausschließlich)
  - Nutzung durch „Nicht-Informatiker“: Erhöhung der Lösungskompetenz „vor Ort“
  - Offenes System, Erweiterbarkeit
- Basis-Eigenschaften
  - Modellierung
  - Zugeschnittene Code-Generierung für eingebettete Systeme
  - Integriertes Debugging
  - Live-Simulation
  - Definierte Schnittstellen für alle Komponenten

# EasyLab - Überblick

- Ansätze
  - Zwei Modellierungs-Paradigmen
    - Synchroner Datenfluss
    - Ereignisbasierter Kontrollfluss
  - Template-basierte Code-Generierung
  - Erweiterbare Software-Bibliothek
- Verwertung
  - Einsatz in der Lehre (berufliche Ausbildung und Hochschulen)
  - Vertrieb in Kombination mit EasyKit-Hardware

# EasyLab - Datenfluss

- Synchroner Datenfluss
- Graphisches Verschalten von Funktionsblöcken
  - Ein- und Ausgänge
  - Interne Zustände
  - Typisierte Verknüpfungen
- Intuitiv, entspricht „Ingenieurs-Denken“ in „Black Boxes“
- Generierung von sicherem und zuverlässigem Code
  - Formalisierung der Modellierung bietet Möglichkeit zur Verifikation
  - Berechnung statische Ausführungspläne
    - Effizient: kein Scheduling zur Laufzeit
    - Berechenbarer Speicherbedarf

# EasyLab - Oberfläche

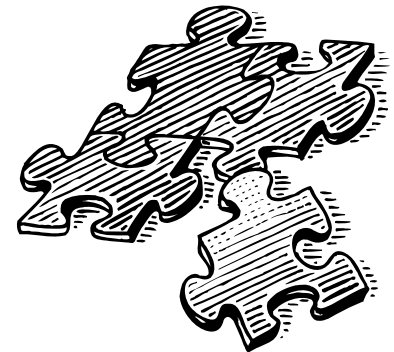
Vergleiche z.B. LabView, Matlab (State-of-the-art in Ingenieurs-Anwendungen)

# EasyLab - Modellierung der Hardware

- Unterstützung unterschiedlicher Prozessorarchitekturen
  - Atmel AVR AT90CAN128
  - Microchip PIC 18F2520 und 18F2580
  - weitere in Vorbereitung
- Plattform-unabhängige Modellierung
- Nahtlose Integration von
  - reinen Software-Funktionsblöcken
  - Hardware-Treiber-Blöcken
- Plattform-Modell
  - Prozessor-Architektur
  - Werkzeugkette (Compiler, Linker, ...)
- Hardware-Modell
  - Beschreibung und Konfiguration der vorliegenden Hardware
  - Gerät: Beschreibung von Match-X-Stapeln
  - Modul
    - Beschreibung von Hardware-Funktionsblöcken und
    - Zuordnung von zugehörigen Softwareblöcken
  - Grundlage für Anbindung von Hardware-Entwurfswerkzeugen (Hardware/Software-CoDesign)

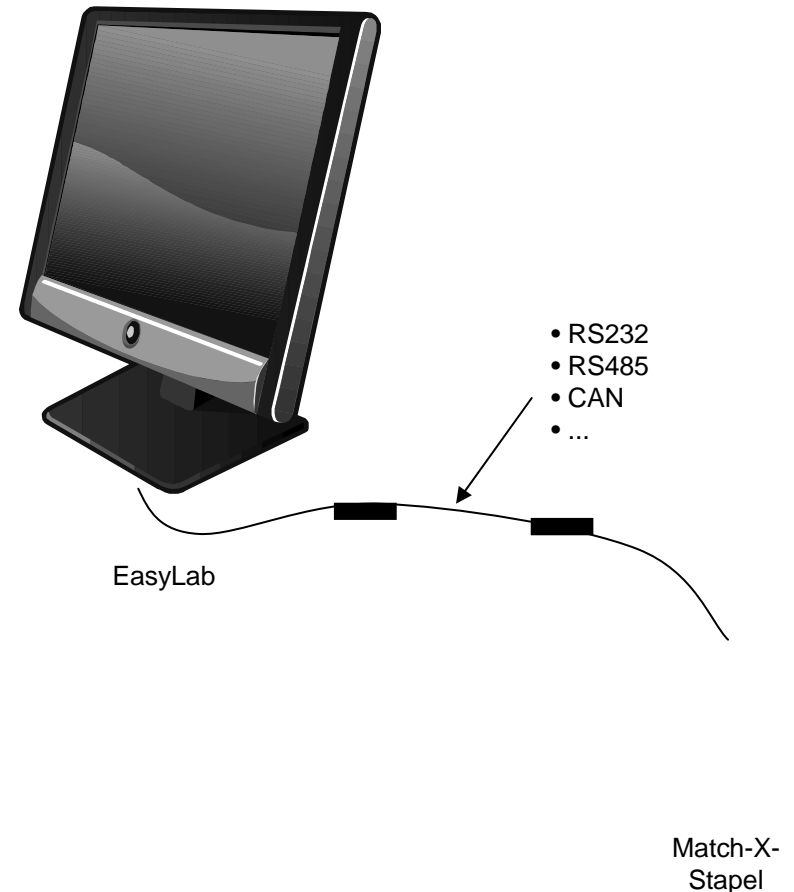
# EasyLab - Modularität

- Definition von neuen Funktionsblöcken durch Benutzer
  - Beschreibung der Schnittstelle nach außen
  - Templates für Code-Generierung
  - Im XML-Format
    - Validierbar mittels Schema: Standardisierung der Software-Schnittstelle
    - Leichte Verarbeitung durch externe Werkzeuge
  - Plugin-Schnittstelle
    - Simulation
    - Visualisierung
  
- Hierarchische Komposition von Funktionsblöcken
  - Gruppierung von verschalteten Funktionsblöcken zu übergeordnetem Block
  - Wiederverwendung von vorhandenen Komponenten



# Parametrisierung und Debugging

- Vom Code-Generator auf Wunsch erzeugt
- Auslesen und Verändern von internen Zuständen
- Übertragung über jeweils verfügbares Medium
- Voraussetzung für Umsetzung didaktischer Konzepte
  - Anschauliche Visualisierung der implementierten Algorithmen
  - Benutzer interagiert über EasyLab mit dem mechatronischen System



# Standardisierung

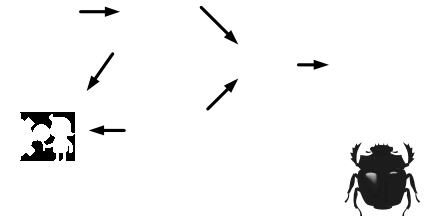
- Bestehender Match-X-Standard
  - VDMA Einheitsblatt 66305
  - Definition u.a. der geometrischen und elektrischen Schnittstelle
  
- Arbeitskreis „Schnittstelle“ (VDMA)
  - Überarbeitung des bestehenden Standards
  - Erweiterung um Software-Schnittstelle
    - Abgeleitet aus modularem Design von EasyLab
    - Ermöglicht nahtlose Integration von eigenen Hardware-Bausteinen in EasyLab

	5	6	7	8	9	10	11	12
E		MS1	MS2	MS3	MS4	RB6	RB7	GD2
F	PRG	RES	CS	RC0	NC	SDO	SDI	UD2
G	RC1	INT	GD1	X	X	UD1	SCK	RC2
H	X	CANL	TXD	X	X	RXD	CANH	X
J			X	X	X	X	X	
K		NC	UA1	PT1	PT2	GA1		
L		NC	URP	PT3	PT4	URM		
M			GA2	NC	NC	UA2		

elektrische Schnittstelle

# Ausblick: EasyKit ist...

- Hardware
  - Modulare Hardware-Plattform
  - Abgeschlossene Entwicklungsmethodik
    - Rapid-Prototyping
    - Übergang zum Serienprodukt unter Erhaltung der Software-Kompatibilität
- Software
  - Modellierung / Entwicklung
  - Integrierte Fehlersuche
  - Simulation
  - Erweiterbarkeit
- Anwendung
  - Verschiedene industrielle Projekte
  - Didaktik (berufliche Ausbildung und Hochschulen)



Vielen Dank  
für Ihre Aufmerksamkeit!