

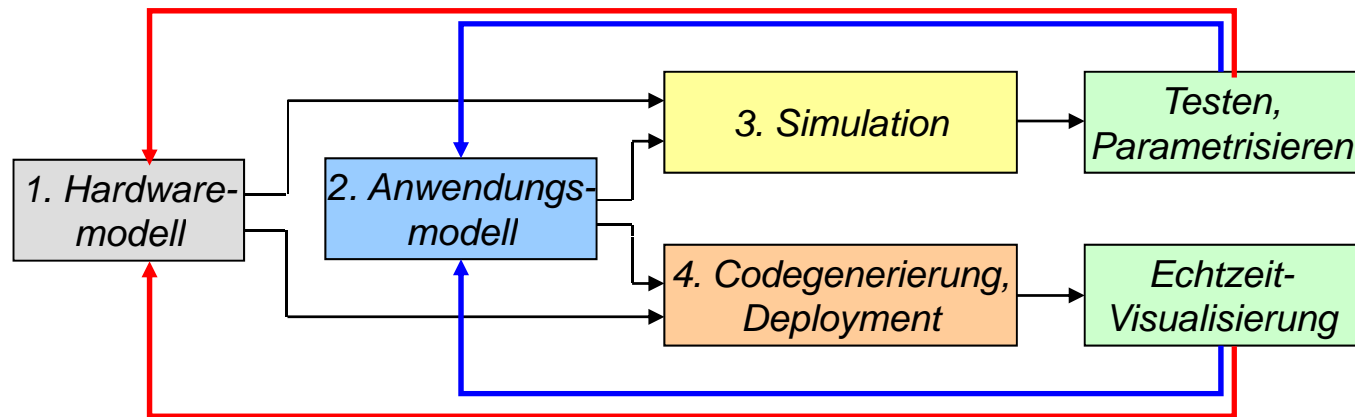
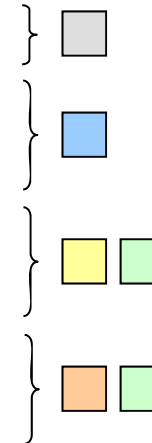
Modellierung von Echtzeitsystemen

Synchroner Datenfluss

Werkzeug: EasyLab

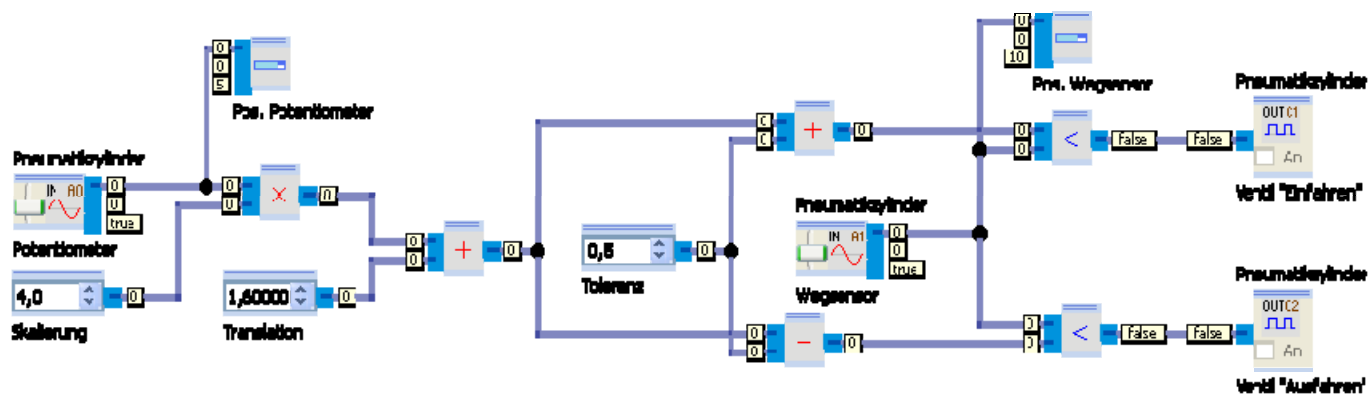
Entwicklungsprozess in EasyLab

1. Spezifikation der Zielhardware
2. Modellierung der Zustandslogik sowie der abzuarbeitenden Aufgabe je Zustand
3. Simulation des Programms zum Testen und zur Erkennung von Fehlern
4. Codegenerierung und Echtzeit-Visualisierung des Zustands der Zielhardware



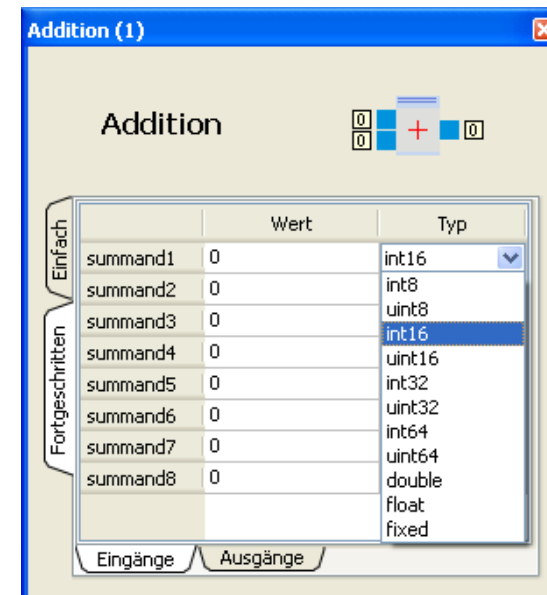
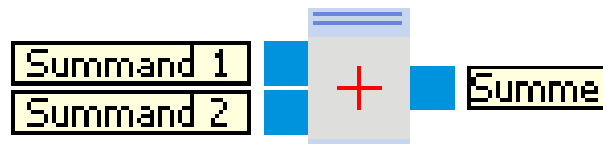
Synchroner Datenfluss

- Funktionsblöcke
 - Eingänge und Ausgänge
 - Interner Zustand
 - Zugeordnete Aktionen
- Komposition von Funktionsblöcken
 - Typisierte Verbindungen
 - In Bearbeitung: Hierarchische Funktionsblöcke
- Grundlagen
 - Synchronitätshypothese
 - „Black boxes“-Sicht
 - Effizienz und Zuverlässigkeit
- Berechnung statischer Schedules
- Deterministisches Laufzeitverhalten
- Statische Speicherallokation (RAM)
- Erzeugung kompakten Codes (ROM)



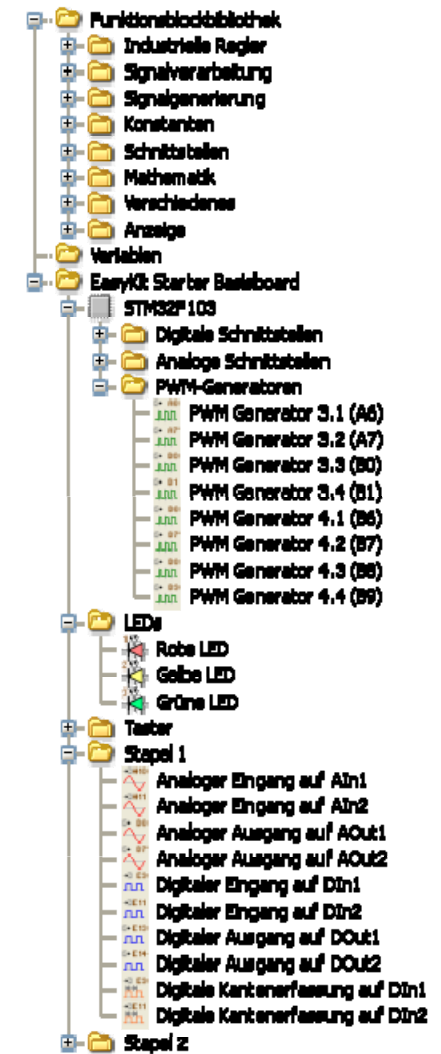
Überladen von Aktoren

- Aktoren können typunabhängig angeboten werden
- Der Typ eines Aktors ist solange frei wählbar, bis sich durch Anschlussbelegung der Typ automatisch ergibt



Geräte

- Hardware
 - Mikrocontroller
 - Sensoren und Aktoren
- Geräte
 - Hierarchische Beschreibung der Hardware
 - Ressourcenmanagement
 - Modellierung der Hardwarefunktionalität
 - Beispiele: I/O-Pins, ADCs, Timer,
 - Schnittstelle zur Anwendungslogik
 - Hardwarezugriff
 - Geräte bieten eine Menge von Funktionsblöcken an



Anwendung – EasyKit Starter

Ansteuerung: Analog

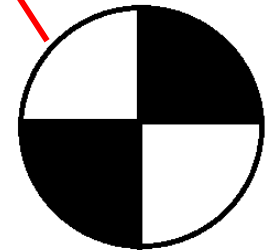
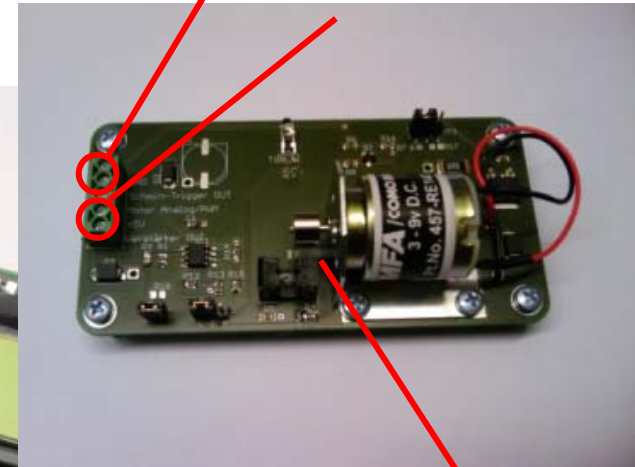
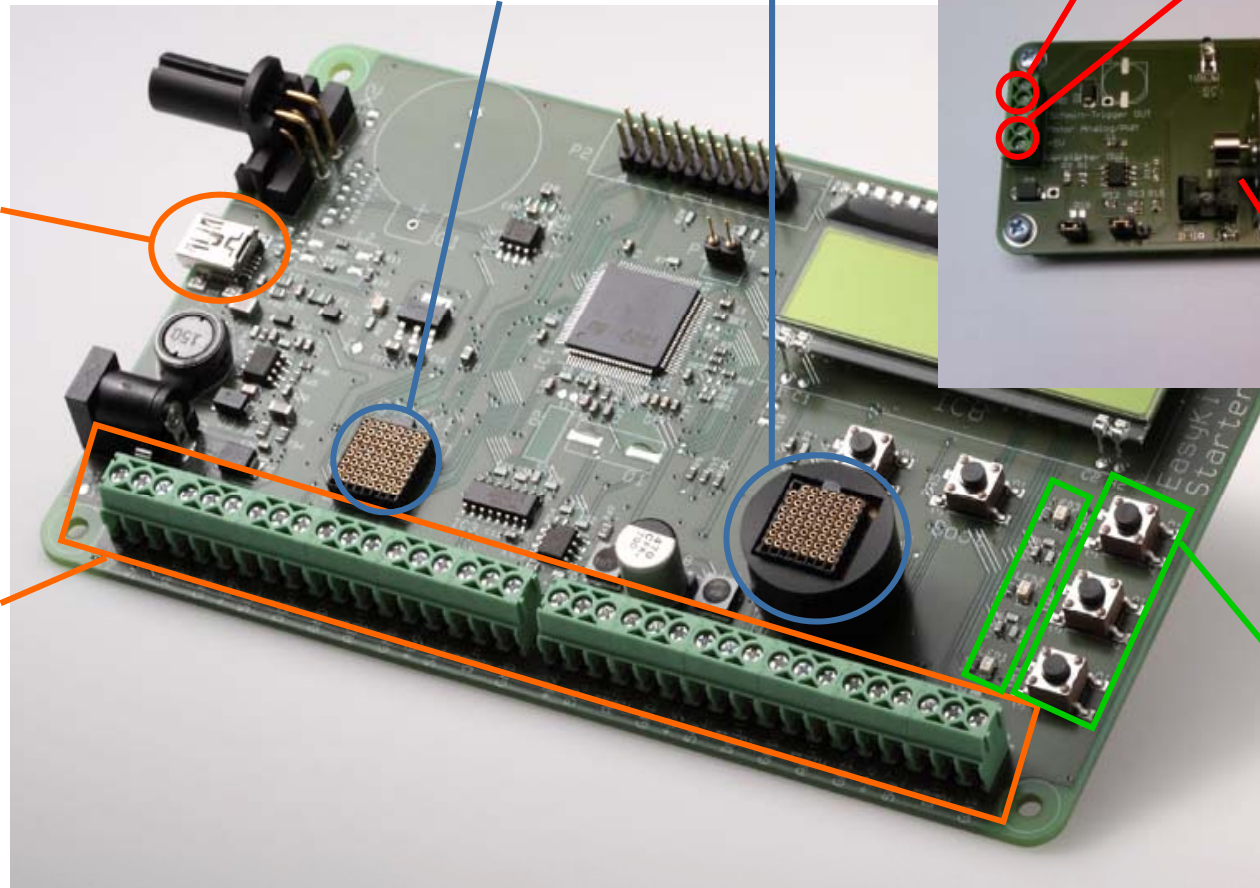
Drehzahl:
Rechtecks-Signal

freier
Sockel

Sockel mit
Match-X-Modul

USB-Anschluss
(Kommunikation,
Stromversorgung)

Anschlüsse für
Sensorik/Aktorik

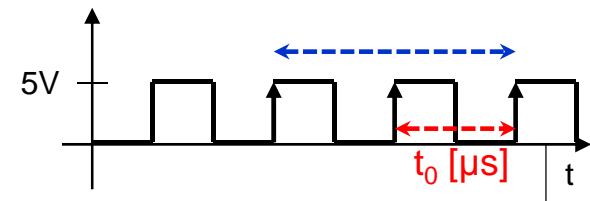
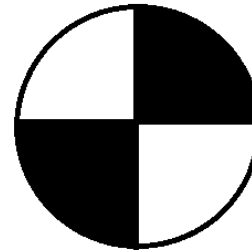


Taster und LEDs
(Interaktion,
Visualisierung)

Aufgabe – Drehzahlregelung

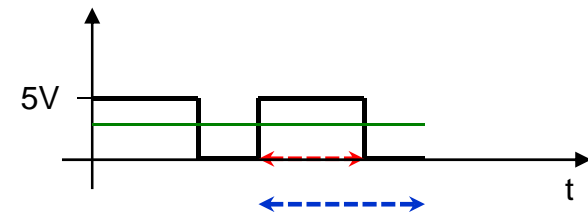
- Eingabe

- Sollwert
- Ist-Wert: Lichtschranke;
Messung der Zeit t_0 zwischen
zwei steigenden Flanken des Signals

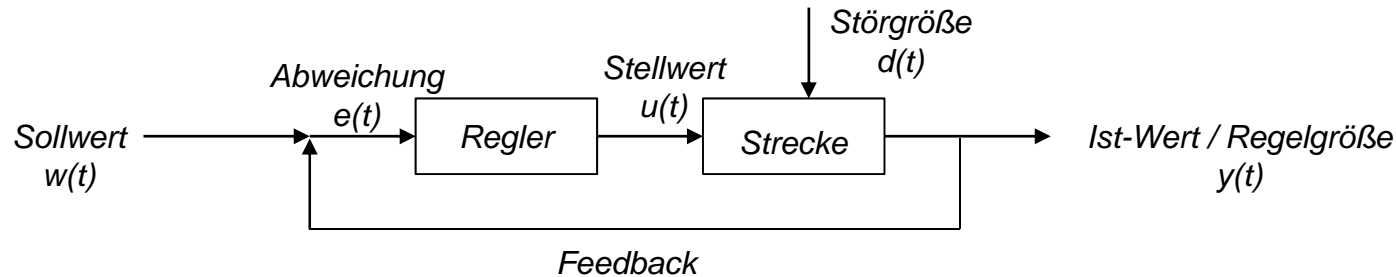


- Ausgabe

- Stellwert: Motorspannung 0..5 V
als Tastverhältnis (in Prozent)



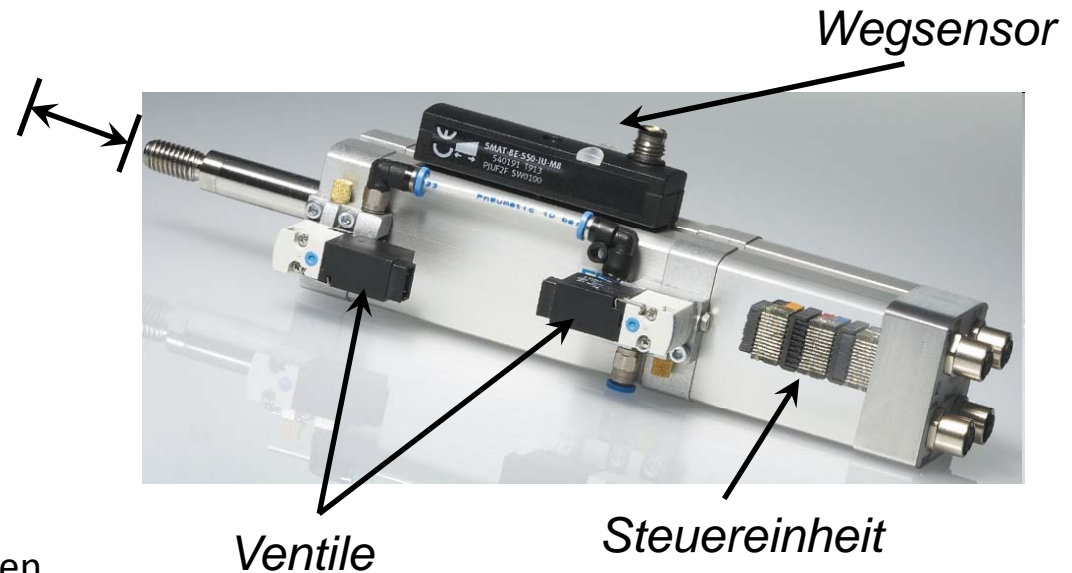
Aufgabe – Drehzahlregelung



- P-Regler $u(t) = K_p \cdot e(t)$
- I-Regler $u(t) = \frac{1}{T_N} \int_0^t (\tau) d\tau$ T_N : Nachstellzeit
- PI-Regler $u(t) = K_p \cdot [e(t) + \frac{1}{T_N} \int_0^t (\tau) d\tau]$

Anwendung – Pneumatischer Zylinder

- Hardware
 - Zylinder
 - Positionssensor (Kolben)
 - Endlagenschalter
 - Zwei Magnetventile
 - Steuerungseinheit
 - Mikrocontroller
 - Analog-Digital-Wandler
 - Treiber für induktive Lasten
- Ziel: Positionssteuerung des Kolbens
- Umsetzung
 - Hardware-Modell aus Bibliothek für Match-X
 - Anwendungsmodell
 - Kleines Datenflussdiagramm
 - Integration der Hardwarefunktionalität



Microchip PIC 18F2520	
efm-systems BSRM (white/black)	
efm-systems CPU (orange/red)	
efm-systems ADU (turquoise/white)	
Analog Input RA0	
Analog Input RA1	
efm-systems PWMD (orange/black)	
PWMD RC1	
efm-systems PWMD (orange/white)	
PWMD RC2	

EasyLab: Zustandsfluss-Diagramme

- Zustandsfluss-Diagramme
 - An IEC-61131-3 angelehnt
 - Zustand: Referenz auf SDF-Modell
 - Transitionsbedingungen: Boolesche Ausdrücke mit Variablen
- Komposition von Zuständen
 - Zustandsfolgen
 - Alternativ- und Parallelzweige
 - Sprünge
- Vorteile
 - Modellierung von Zuständen (vgl. Automaten)
 - Diagrammart weit verbreitet in Anwendungsdomäne

